



# Restoration: Image Inpainting Using Global Image Mathematics

Rajesh A. Jadav<sup>1,\*</sup>, Dilip C. Joshi<sup>2</sup> and Shailesh S. Patel<sup>3</sup>

<sup>1</sup>Department of Mathematics, S.V.M. Institute of Technology, Bharuch, Gujarat. India.

<sup>2</sup>Department of Mathematics, Veer Narmad South Gujarat University, Surat, Gujarat. India.

<sup>3</sup>Department of Mathematics, GIDC Engineering College, Navsari, Gujarat. India.

## ARTICLE INFO

### Article history:

Received: 3 March 2014;

Received in revised form:

30 March 2014;

Accepted: 17 April 2014;

### Keywords

Image Inpainting,  
Image Restoration,  
Diffusion Equation,  
Partial differential equation.

## ABSTRACT

Image Inpainting and denoising refers to the ill-posed problem of filling in the missing data in digital images by interpolating from the vicinity. In this paper, we present a new approach for image restoration. The denoising is performed by the smoothing equation working inside and outside of the inpainting domain but in completely different ways. Inside the inpainting domain, the smoothing is carried out by the Mean Curvature Flow, while outside the inpainting domain is carried out in a way as to encourage smoothing within a region and discourage smoothing across boundaries. The approach here presented permits the transportation of available information from the outside towards the inside of the inpainting domain. To illustrate the effective performance of our model, we present some experimental results on a Matlab and solved the diffusion equations.

© 2014 Elixir All rights reserved.

## Introduction

Inpainting, dis-occlusion and filling-in are various names for the same task. Given an image with a missing region, i.e. a hole, restore the values in the hole in an undetectable way [5]. The inpainting problem is clearly ill-posed. Any method must therefore use some prior assumptions about the unknown missing values and their relations with the known hole neighbourhood. Applications include restoration of photographs, films and paintings, text and objects removal in images, special effects in movies, disocclusion in computer vision.

Inpainting was initially introduced into image processing by Bertalmio et al [6]. The authors used partial differential equations (PDE) method [1,2,4] to restore images. Their algorithm fills in the areas to be inpainted by smoothly propagating information from the surrounding areas along the isophote direction. It does a good job in small damaged regions, but for larger areas, the results usually look blurry. Lately, Chan and Shen proposed two inpainting models: the Total Variation (TV) [7] and the Curvature-Driven Diffusion (CDD) models [8]. They converted the problem into finding out the extrema of energy functions. However, their models still only aim at handling local non-texture inpainting [7,8]. The difficulty of real inpainting problems is due to the rapid variations of isophote and the roughness of image functions. Besides, exemplar-based and region-filling methods [11,12] search the optimal matching patches within the valid image regions, and fill them to the damaged areas according to the order of repair priority. Experiments show that these methods can effectively fill the larger damaged image areas, but the search process is very slow and prone to false match. So in this paper we developed the new task for the image restoration, generally the inpainting algorithms is to filling-in regions with available information from their surroundings. In most cases, the available data of the original image is noisy which makes it necessary to eliminate the noise and filling-in the blank spaces (those without information). The basic idea of our algorithm is to complete these spaces which hold no information and eliminate noise (if exists) while preserving the edges, and the goal of this work is to recover the entire clean image  $u(x)$  from a given incomplete noisy image  $I(x)$  observed only outside of an inpainting domain  $D$ , performing the inpainting and denoising action simultaneously.

To well recognize the proper Image inpainting [5] and denoising techniques applicable to image restoration, it is necessary to have a good understanding of inpainting as well as denoising problems. Here, we first give a generalized impact of global image mathematics [2,3] and fundamental of image inpainting problems.

### Global Image Mathematics in Image Inpainting:

Inpainting problem defined mathematically, in [5] as a sequence say  $S$ , given only a subsequence of it,  $X$  estimate the whole  $S$  as  $S'$  such that  $I(S') = I(X)$ , where  $I$  denotes the information. Here is a simple example to explain it more clearly. Suppose there is a sequence  $\{1, 2, 3, 4, X, 6\}$  where,  $X$  is the unknown element. If  $X$  is derived as 5, the whole sequence looks very “natural”, for  $I(X = 5 | \{1, 2, 3, 4, X, 6\}) = 0$  i.e., it takes the exact value as we expected. However, if  $X$  is derived as 10, then the whole sequence does tell us something unexpected, and  $I(X = 5 | \{1, 2, 3, 4, X, 6\}) > 0$ . In case of inpainting, the generated plausible regions are commonly looks natural which indicates that no additional information can be reproduced out of nothing related. So, now we define general principle and fundamental of Image inpainting for the image restoration task.

### Fundamental of Image Inpainting

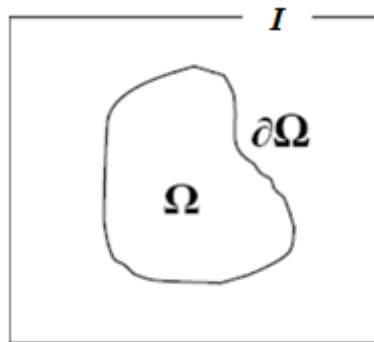


Fig.1: Image  $I$ , the region  $\Omega$  to be inpainted and its boundary  $\partial\Omega$ .

Digital inpainting refers, as already mentioned, to inpainting through some sort of image restoration. The digital inpainting process can be looked upon as a linear or non-linear transformation, where,  $I$  is the original image and  $u$  is the transformed image. e.g.  $I \rightarrow \text{Image Processor } f \rightarrow u$ . In mathematically, the image processor can be looked upon as a function  $f: I \rightarrow u$  i.e.  $u = f(I)$ .

General Principle: Let  $\Omega \subset I$  stands for the region to be inpainted as shown in fig.1, and  $\partial\Omega$  for its boundary. The objective is to fill the hole  $\Omega$  with appropriate gray values by interpolating the data located at the neighbourhood in the surrounding region  $\partial\Omega$ .

The rest of the paper is organized as follows. In Section II, we briefly describe the inpainting scheme given in [6] and the Mean Curvature Flow model. In section-III, we describe the proposed model for inpainting and denoising. The Euler discretization of the proposed model and the numerical implementation in Matlab, are discussed in Section IV. And concluding remarks are presented in Section V.

### Description of Image Inpainting and Denoising:

Inpainting is a practice carried out by artists when modifying a picture, in such a way so that an observer is unable to detect any changes. The goal of the inpainting model introduced in [6] is to transport as smoothly as possible (along the isophotes) the information from the surrounding inpainting domain. The transport of information is performed by solving the following partial differential equation

$$\frac{\partial u}{\partial t} = \nabla(\Delta u) \cdot \nabla^\perp u, x \in D, t > 0 \dots \dots \dots (1)$$

where,  $t$  is a scale space parameter as in [6,8]. Note that this evolution equation runs only inside the region to be inpainted  $D$ . In Eq. (1),  $\nabla^\perp u$  is a vector which indicates the direction of  $u(x)$  variation of least intensity. The absolute value of  $\nabla^\perp u$  is numerically equal to the instant variation rate of  $u(x)$  along the isophotes. The vector  $\nabla(\Delta u)$  indicates the direction where the value of  $u(x)$  varies abruptly (indicating a border or edge) and its absolute value is numerically equal to the laplacian instant variation rate

in this direction. The inner product of these two vectors gives us the value to be transported to the pixel in question. We observe that in the case where the two vectors are perpendicular the transport will not be carried out.

The inpainting method proposed in [6,14] consists of intercalating the Eq. (1) with a diffusion equation whose objective is, to ensure a correct evolution of the direction field [6]. They used the following anisotropic diffusion equation

$$\frac{\partial u}{\partial t} = g_{\varepsilon} K |\nabla u|, x \in D^{\varepsilon} \dots\dots\dots (2)$$

where,  $D^{\varepsilon}$  is a dilation of  $D$  with radius  $\varepsilon$ ,  $K$  is the Euclidean curvature of  $u$ , and  $g_{\varepsilon}$  is a smooth function in  $D^{\varepsilon}$  such that  $g_{\varepsilon} = 0$  in  $\partial D_{\varepsilon}$  and  $g_{\varepsilon} = 1$  in  $D^{\varepsilon}$ .

Several PDE-based techniques have been proposed for the smoothing of an image, some obtained on the direct derivation of the evolution equations, and others from energy approaches such as the  $L^2$  norm dependent model or from the total variation model of Rudin, Osher and Fatemi [16]. Here, we will smooth the data inside the inpainting domain using the Mean Curvature Flow (MCF) equation (see [6,14] for further details), which is given by

$$\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) \dots\dots\dots (3)$$

This equation diffuses  $u$  in the direction orthogonal to its gradient  $\nabla u$  and does not diffuse in the direction of  $\nabla u$ . The level sets of the solution of the MCF equation move in the normal direction with a speed proportional to their mean curvature.

#### The proposed Method for Improvement of Inpainting and Denoising problem:

The proposed method in this work will deal with inpainting task is to reconstruct a damaged image from an image. Also our method will inpaint the domain  $D$ , which has circular noise [10] or scratches or missing patches as shown in fig. 2(c) and fig. 3(c), and will denoise the image in  $D^C$ . The goal of this work is to develop the idea of removing noise without losing the boundaries or edges and inpaint the damaged regions. Our approach presents different procedures inside and outside of the inpainting domain. The inpainting process consists of filling-in the missing information in the empty domain  $D$ , based upon the available image information found outside of the domain  $D$  as in [14,15], and will be performed only inside the inpaint domain  $D$  while the smoothing procedure will be performed in all  $X \in \Omega$  but will act differently depending on if point  $x$  belongs, or not, to the inpainting domain  $D$ .



Fig. 2:- (a) Original Image; (b) Mask; (c) Imposed Mask (Circular mask) to Original Image



Fig. 3:- (a) Original Image; (b) Mask; (c) Imposed Mask (Circular mask) to Original Image

**Inpainting:** The algorithm will execute, for each  $x$  belonging to the inpainting domain  $D$ , the transportation of neighbouring pixel information belonging to  $D^c$ , where the image information is available, using the Eq. (1) as in [14].

**Smoothing:** A smoothing procedure will be applied for each  $x$  belonging to the domain  $\Omega$ . If  $x \in \Omega$ , the smoothed  $u(x)$  will be the solution of the Mean Curvature Flow equation, otherwise, i.e. if  $x$  lies outside of the inpainting domain a selective diffusion equation will be used. The smoothed version  $u(x)$  will be the solution to the following parabolic equation

$$\frac{\partial u}{\partial t} = g|\nabla u| \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right) - (1-g)(u-I), x \in D^c, t > 0. \dots\dots\dots(4)$$

$$\frac{\partial u}{\partial t} = |\nabla u| \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right), x \in D. \dots\dots\dots(5)$$

$$u(x,0) = I(x), x \in D (\text{e.g. Initial Condition}) \dots\dots\dots(6)$$

$$\frac{\partial u}{\partial t} : \partial\Omega \times R_+ = 0, x \in \partial\Omega, t > 0, (\text{reflecting Boundary Condition}) \dots\dots\dots(7)$$

$$\text{where, } g = g(|\nabla u|)$$

Here  $I(x, y)$  is an image to be processed,  $u(x, y, t)$  is its smoothed version on the scale  $t$ . The function  $g(s) \geq 0$  is a non-increasing function, satisfying  $g(0) = 1$  and  $g(s) \rightarrow 0$  when  $s \rightarrow \infty$ . The term  $|\nabla u| \operatorname{div}(\nabla u/|\nabla u|) = \Delta u - \nabla^2 u(\nabla u, \nabla u)/|\nabla u|^2$  diffuses  $u$  in the orthogonal direction to its gradient  $\nabla u$  and does not diffuse it in any other direction. The goal is to allow smoothing in the image  $u$  in a way that it is performed on both sides of an edge with minimal smoothing on the edge itself. The term  $g(|\nabla u|)$  is used for edge detection and controls the diffusion speed. If  $\nabla u$  has a small value at the point  $x$ , it will be considered an interior point, and the diffusion will be stronger. On the other hand, that is, if  $\nabla u$  has a large value at the point  $x$ , then this point  $x$  will be considered an edge point, and the diffusion will be low since  $g(s)$  always assumes values for large values of  $s$ . The balance between the forcing term and the diffusion term is made by  $(1-g)$ , which works as a moderate selector of the diffusion process. Thus, the proposed model consists of selectively applying the Alvarez, Lions and Morel model's [7] in areas of the image that demand a larger suavization. This model also consists of forcing, in an incisive way, the smoothed image  $u$  to remain close to the initial image  $I$  in the boundary areas which have  $g \sim 0$ . On the other hand, in homogeneous areas  $g \sim 1$ , and therefore, the forcing term will have an inexpressive effect, which allows for a better suavization of the image.

The non-linear space scale scheme (4), (6) and (7) for data smoothing was introduced in [11] following the ideas given in [7,8,13], with the objective of eliminating unnecessary parameters, where in many cases they are correlated. This equation allows one to perform selective smoothing in accordance with the value of the image gradient at each point  $x$ . We would also like to mention some other interesting works on inpainting. In [7,14], the authors present a technique for removing occlusion solving the following variational formulation

$$E(u) = \int_D |\nabla u| \left( \alpha + \beta \left| \operatorname{div} \frac{\nabla u}{|\nabla u|} \right|^p \right), p \geq 1, \alpha > 0, \beta > 0 \dots\dots\dots(8)$$

to perform disocclusion and the practical algorithm for disocclusion, connecting the T-junctions associated with the same gray level using elastica minimizing curves. As reported by the authors, the regions to be inpainted are limited to having simple topologies, e.g. holes are not possible taking into account the specific numerical technique proposed. In [13,14], the authors work on the minimization of the elastica energy functional (8) obtaining the Euler–Lagrange equation for this functional. The choice of values for constants  $\alpha$

and  $\beta$  is an issue of numerical implementation. There has been no investigation about the range of these two constants, which permits convergence and/or stability of the numerical algorithm.

#### Numerical approximation and Implementation results in Matlab:

Here, images are represented by  $256 \times 256$  matrices of intensity values, where each matrix element  $u_{i,j}$  is a real value correspondent to the gray scale level of the image  $u(x, y)$  at the point  $x = x_i = i\Delta x$  and  $y = y_i = i\Delta y$ . We denote  $u(x_i, y_i, t_n)$  by  $u_{i,j}^n$ , where  $t_n = n\Delta t$ . The derivative of  $u$  in relation to the time  $t$  i.e.  $\frac{\partial u}{\partial t}$  calculated in  $(x_i, y_j, t_n)$  is approximated by Euler's

$$\text{method, i.e. } \frac{\partial u}{\partial t} \approx \frac{(u_{i,j}^{n+1} - u_{i,j}^n)}{\Delta t}.$$

Also, Numerical implementation using central difference techniques [9] works quite well in the approximation of the diffusion term  $|\nabla u| \left( \text{div} \left( \frac{\nabla u}{|\nabla u|} \right) \right) = \frac{u_x^2 u_{yy} - 2u_x u_y u_{xy} + u_y^2 u_{xx}}{u_x^2 + u_y^2}$ , and the transport term  $\nabla(\Delta u) \cdot \nabla^\perp u = u_x(u_{xxy} + u_{yyy}) - u_y(u_{xxx} + u_{xyy})$ .

Using Neumann's boundary conditions [4,9] we calculate  $u_{ij}^{n+1}, n = 1, 2, \dots, N$ , by the discretization of the (1), (4) and (5), we have

$$v_{i,j}^{n+1} = u_{i,j}^n + \Delta t_i L_i(u_{ij}^n), (x, y) \in D$$

$$v_{ij}^{n+1} = u_{ij}^n, (x, y) \in D^c$$

$$u_{i,j}^{n+1} = v_{i,j}^{n+1} + \Delta t_s L_s(v_{ij}^{n+1}), (x, y) \in \Omega$$

with  $u_{ij}^0 = I(x_i, y_i), L_i(u) = \nabla(\Delta u) \cdot \nabla^\perp u$  and

$$L_s(v) = g|\nabla v| \text{div} \left( \frac{\nabla v}{|\nabla v|} \right) - (1-g)(v-I), \text{ if } (x, y) \in D^c, \text{ or}$$

$$L_s(v) = |\nabla v| \text{div} \left( \frac{\nabla v}{|\nabla v|} \right), \text{ if } (x, y) \in D$$

here,  $g = g(s) = 1/(1 + ks^2)$ , where  $k$  is a constant and  $s = |\nabla v|$ .

The computational code was written in Matlab. The results were obtained by using Intel(R) Core(TM) 2 Duo (T6500 @ 2.10 GHz, 4 GB RAM). The running time for an image of  $256 \times 256$  pixels size is about 8 to 10 second with less than 100 iterations.

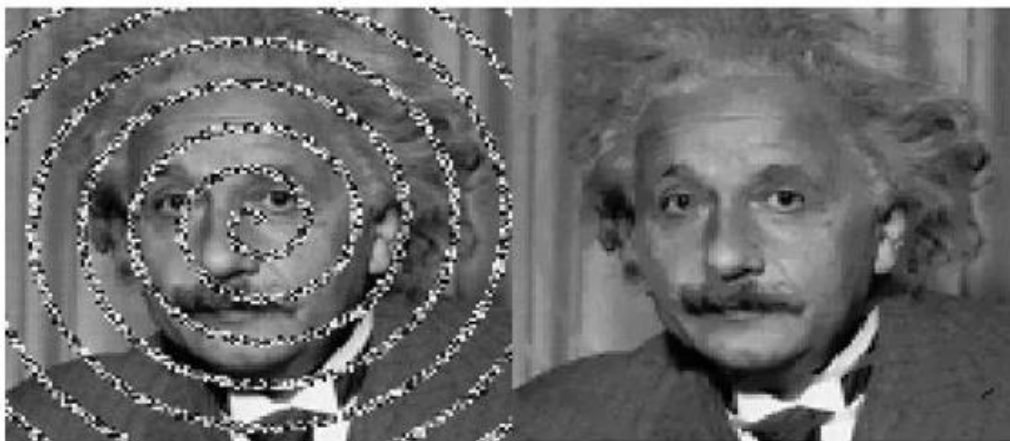


Fig. 2.1:- (a) Circular mask Noisy Image; (b) Recovered Inpainted Image

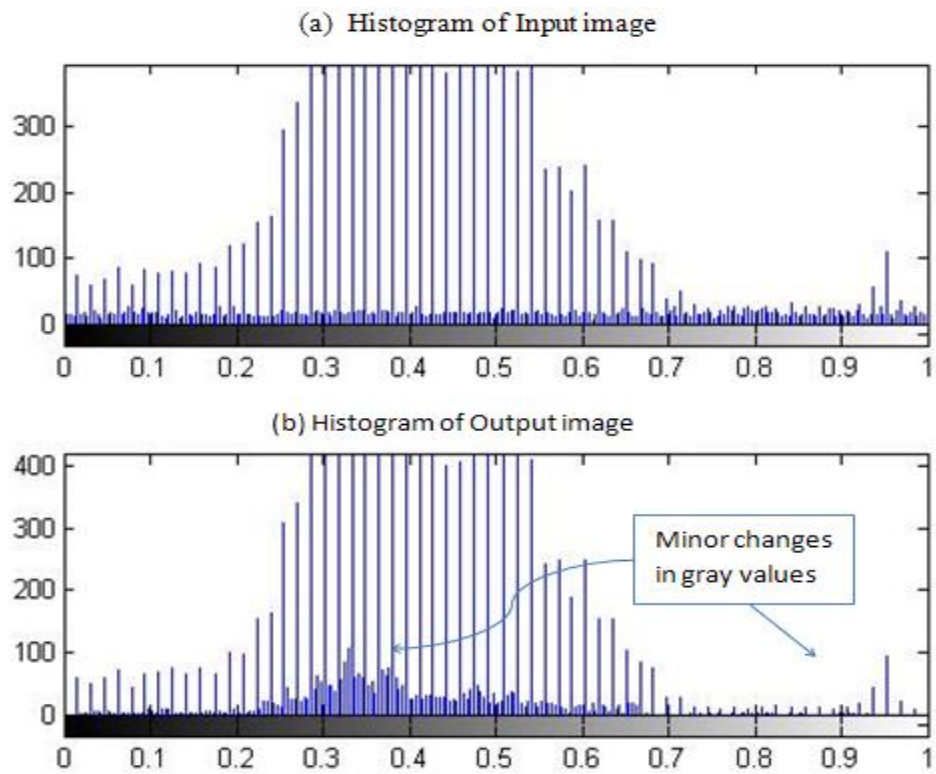


Fig. 2.2:- (a) Histogram of Input Image; (b) Histogram of Inpainted (output) Image

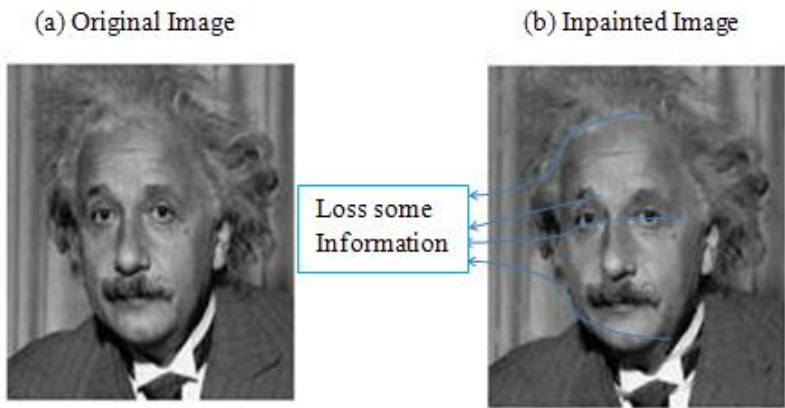


Fig. 2.3:- Comparison: - (a) Original Existing Image; (b) Recovered Inpainted Image

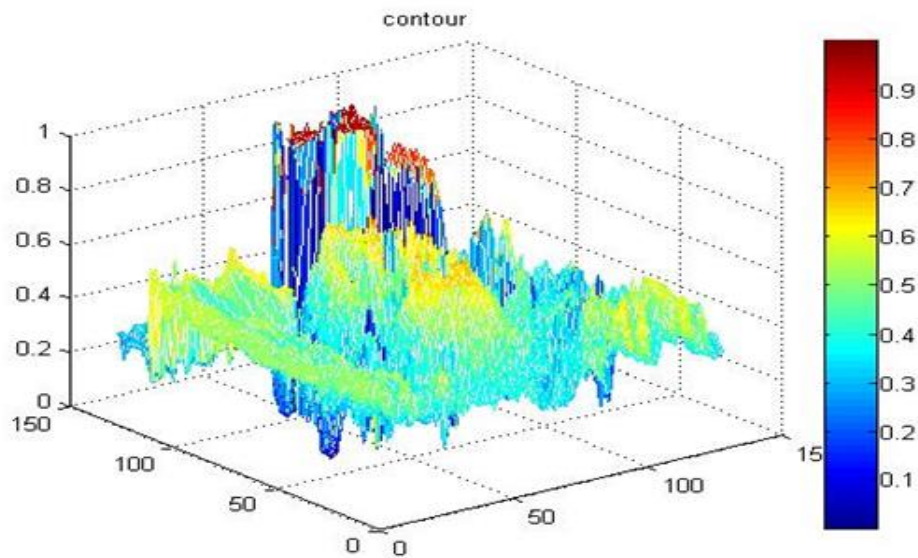


Fig. 2.4:- Contour plot of Inpainted (Recovered) Image



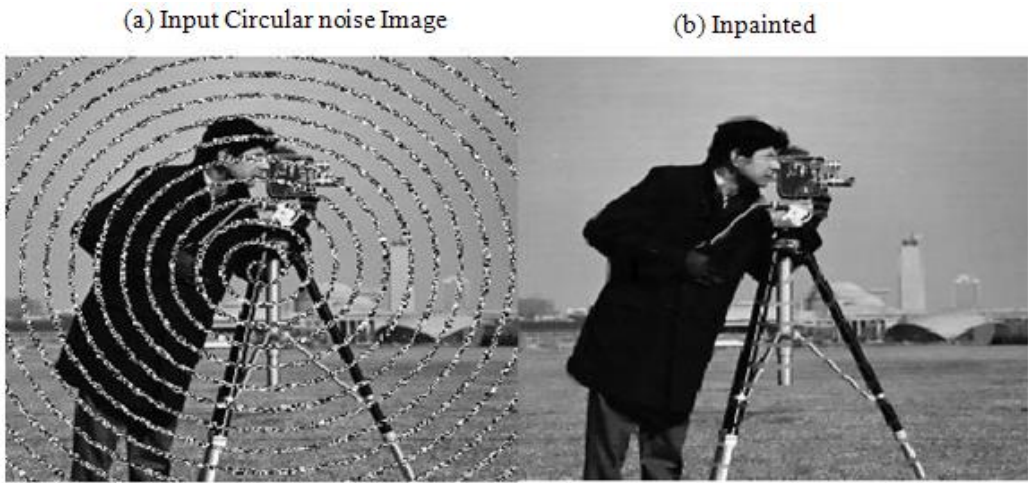


Fig. 3.1:- (a) Circular mask Noisy Image; (b) Recovered Inpainted Image

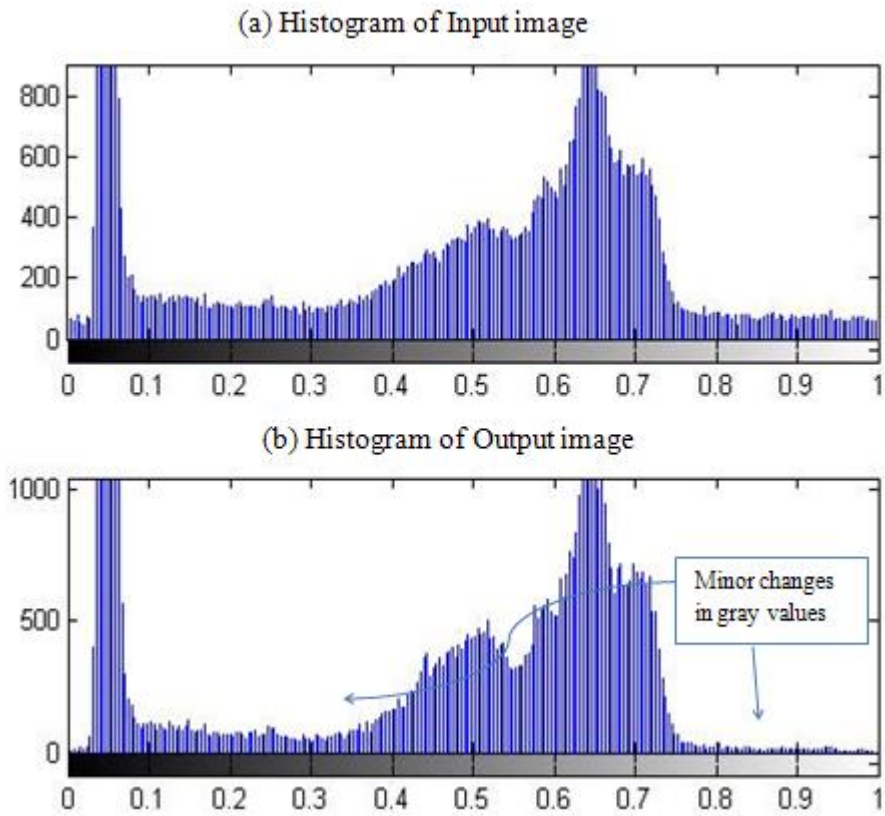


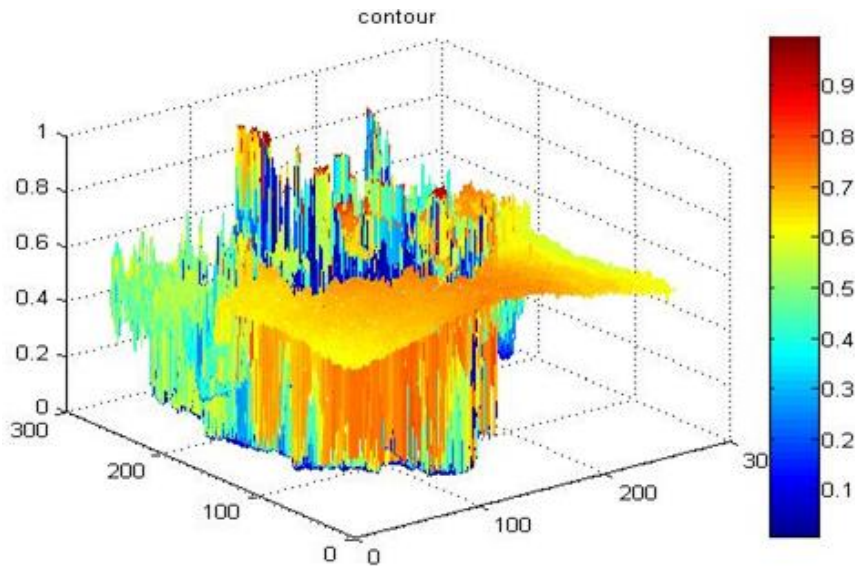
Fig. 3.2:- (a) Histogram of Input Image; (b) Histogram of Inpainted (output) Image



Fig. 3.3:- Comparison: - (a) Original Existing Image; (b) Recovered Inpainted Image

The preceding results illustrate the performance obtained by the proposed model. We performed inpainting task on typical two images and as the principal contribution of the proposed model is the case where the initial image  $I(x)$  was damaged by circular noise, also depending on user choice we may include a region with lost or no information data. So, here we describe the parallel representation of two input images with the experimental result done in Matlab.

Here we consider fig. 2(a) and fig. 3(a) are original images and fig. 2(c) and fig. 3(c) are circular noisy (mask) images. When we applied our proposed method to the given problem then circular noise is removed within less than 100 iterations and we have recovered image looks like an exactly original image, i.e. we called the inpainted image as shown in fig. 2.1 and fig. 3.1. Also with the help of histogram we can verify our existing result is correct or not.



**Fig. 3.4:- Contour plot of Inpainted (Recovered) Image**

So that we have the fig. 2.2 and fig. 3.3 that shows the histogram of original image and inpainted image respectively. The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. More, generally we say histogram shows the distribution of data values. So our result tells us that there is no any major difference of gray level occurs between input and output of the images. At last we also find the contour plot of the recovered image i.e. Inpainted image, which is shown in fig. 2.4 and fig. 3.4 respectively.

#### **Concluding Remarks:**

In this paper, we present a mathematical model for image restoration, which consists of the recovery of missing parts in an image and also eliminating the corruption caused by noise. The results presented in our examples demonstrate the high performance of the proposed model which has demonstrated great efficiency in dealing with the inpainting of damaged images and denoising when the initial image is noisy. The size of the inpainting domain can interfere with the results. The drawback of the proposed model can be seen when we analyse the image, which is characterized by texture, a great part of which is eliminated, because the inpainting domain is quite large. From the numerical point of view, the number of intercalated steps of inpainting and diffusion should be investigated mainly when the image to be restored has several different inpainting domains.

#### **Acknowledgment:**

Much thanks to my guides for his constructive criticism, and assistance towards the successful completion of this work.

#### **References:**

- [1] A.K. Jain, "Fundamental of Digital Image Processing", *Pearson Education*, India, 2004.
- [2] R.C. Gonzalez, R.E. Woods, S.L. Eddins, "Digital Image Processing using Matlab", *Pearson Education*, India, 2005.
- [3] B.Chanda, D.D.Majumder, "Digital Image Processing and Analysis", *Prentice-Hall* of India, 2002.
- [4] K. Sankara Rao, "Introduction to Partial Differential Equations", *Prentice-Hall* of India, 1995.



- [5] R.A. Jadav, S.S. Patel, D.C. Joshi, “A Comparative study of different techniques in image Inpainting”, *Indian Journal of Engineering*, Volume 3, Number 7, May 2013, pp. 41-44.
- [6] M.Bertalmio, G.Sapiro, V.Caselles, C.Ballester, Image inpainting, *SIGGRAPH 34* (2000), pp. 417-424.
- [7] T.Chan, J.Shen, “Mathematical models for local non-texture inpainting”, *SIAM, Journal of Applied Mathematics*, 2001, 62(3), pp. 1019-1043.
- [8] T.Chan, J.Shen, “Non-texture inpainting by curvature-driven diffusions (CCD)”, *Journal of Visual Communication and Image Representation*, 2001, 12 (4), pp. 436-449.
- [9] R.A. Jadav, S.S. Patel, D.C. Joshi, “Image segmentation based on chen-tese active contours using Finite difference scheme”, *International journal of Scientific-Research and development*, Vol. I, 2013, pp. 1280-1284.
- [10] R.A. Jadav, S.S. Patel, D.C. Joshi, “Image segmentation using active contours using on noisy images: Mathematical approaches”, *Asian Academic Research Journal of Social Science & Humanities*, Vol. I, December 2013, pp. 13-22.
- [11] I.Drori, C.Daniel, Y.Hezy, “Fragment based image completion”, *ACM Trans. on Graphics*, 2003, 22(3), pp. 303-312.
- [12] A.Criminisi, P.Perez, K.Toyama, “Region filling and object removal by exemplar-based image inpainting”, *IEEE Trans. On Image Processing*, 2004, 13(9), pp. 1200-1212.
- [13] J. Malik, P. Perona, “Scale-space and edge detection using anisotropic diffusion”, *IEEE TPAMI 12* (7), (1990), pp. 629–639.
- [14] M. Bertalmio, “Processing of Flat and non-Flat Image Information on Arbitrary Manifolds Using Partial Differential Equations”, *Thesis, University of Minnesota*, 2001.
- [15] L. Alvarez, P.L. Lions, J.M. Morel, “Image selective smoothing and edge detection by nonlinear diffusion”, *SIAM J. Numer. Anal.* 29 (3) (1992), pp. 845–866.
- [16] L. Rudin, S. Osher, E. Fatemi, “Nonlinear total variation based noise removal algorithms”, *Physica D 60* ,(1992). pp. 259–268.