Modified Finite Difference and Linear Iterative PDE Methods in Digital Image Inpainting

Sumudu Kalubowila
Department of Mathematics, Prairie View A & M University

Abstract

Image inpainting process is used to develop the damaged image or missing part of the image. This technique has more applications, such as text removal in the image, photo restoration and etc. There are different methods used in image inpainting, such as nonlinear partial differential equations, wavelet transformation, framelet transformation, etc. In this study a linear diffusion PDE method for image inpainting is considered. And to solve this linear PDE a numerical method was developed. Also, different diffusion conductivity, such as constant and non-constant, were considered for this method. Linear diffusion PDE method was compared with existing non-linear diffusion PDE methods. For an any inpainting method, there exists an error associated with it. So, two different methods were considered to find a relationship between error and inpainting domain.

1. Introduction

Inpainting has been carried out by professional artists for many years. When done manually, it is a very time-consuming process. The basic idea of this process is to reconstruct damaged parts or missing parts of an image. It has important value in restoration of old photographs; the removal of artifacts in a film; the removal of red eye; the removal of superimposed text; and the removal redundant objects etc. In 2000 SIGGRAPH conference, the idea of digital inpainting was established by Bertalmio-Sapiro-Caselles-Ballester [1]. Image inpainting has been expanding very fast. It is a very important topic in the field of Digital Image Processing.

Nowadays data exchange has become popular. Since time and skill are required to do image inpainting manually, it is important to find an automatic and fast method. Therefore, different type of successful inpainting techniques were developed in last few years. The idea of the computer algorithm of image inpainting is to fill these missing data with known data surrounding D. D is the inpainting domain.

2. Background

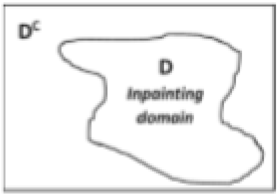

Figure 1. D^c is the known data and D is the inpainting domain

2. Background

2.1. Finite difference method

Laplace equation with Dirichlet boundary condition was considered.

$$\nabla^2 u(z) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad u \in D$$
$$u = f \quad on \quad \partial D$$

Now consider the square domain with boundary conditions;

$$Au = F$$

Where A is a invertible square matrix. Therefore, value of ui is given by,

$$u = A^{-1}F$$

Also, A is a block matrix. So, we can rewrite this matrix as,

$$A = \begin{bmatrix} B & I & O \\ I & B & I \\ O & I & B \end{bmatrix}$$

where,

$$B = \begin{bmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2.2. Nonlinear diffusion PDE

Here we consider the PDE with Neumann boundary condition.

$$\frac{\partial u}{\partial t} = -f(u), \quad in \quad D \quad t \geq 0,$$
$$u(z,0) = u_0(z), \quad z \quad in \quad D,$$
$$\left.\frac{\partial u}{\partial \mathbf{n}}\right|_{\partial D} = 0,$$

Here n is the unit inner normal vector. Therefore, numerical solution of this PDE is,

$$u^{(k+1)} = u^{(k)} - \tau_k f(u^{(k)}) \quad k = 0, 1, 2, 3, ... \tag{2.2.1}$$

where τ_k is the step size.

Now we consider the heat diffusion equation with heat conductivity $c(|\bigtriangledown u|)$.

$$\frac{\partial}{\partial t} u = \bigtriangledown \cdot (c(|\bigtriangledown u|) \bigtriangledown u) \quad in\ D, t \geq 0$$
$$\left.\frac{\partial}{\partial \mathbf{n}} u\right|_{\partial d} = 0$$
$$u(z,0) = u_0(z), \qquad z \in D$$

Now apply the anisotropic diffusion equation [3],

$$\bigtriangledown \cdot (c(|\bigtriangledown u|) \bigtriangledown u) = \bigtriangledown (c(|\bigtriangledown u|) \cdot \bigtriangledown u + c(|\bigtriangledown u|) \Delta u$$

$u_0(z)$ is given by the solution of this PDE,

$$\bigtriangledown^2 u(z) = 0 \quad u \in D$$
$$u = f \quad on \quad \partial D$$

Now we are going to solve this non-linear diffusion PDE using equation(2.2.1).

Therefore,

$$u^{(k+1)} = u^{(k)} - \tau_k \bigtriangledown \cdot (c(|\bigtriangledown u^{(k)}|) \bigtriangledown u^{(k)})$$

Where, k=0, 1, 2, 3, ...

$$u^{(1)} = u^{(0)} - \tau_0 \bigtriangledown \cdot (c(|\bigtriangledown u^{(0)}|) \bigtriangledown u^{(0)})$$
$$where \quad u^{(0)} = u_0$$
$$u^{(2)} = u^{(1)} - \tau_1 \bigtriangledown \cdot (c(|\bigtriangledown u^{(1)}|) \bigtriangledown u^{(1)})$$
$$u^{(3)} = u^{(2)} - \tau_2 \bigtriangledown \cdot (c(|\bigtriangledown u^{(2)}|) \bigtriangledown u^{(2)})$$
$$....$$

We can continue this process and we can get a value of u.

3. Modified finite difference method

3.1. Initial value

When we find the initial value of the inpainting domain, we use five point stencil method. So, in our inpainting domain has m rows and n columns. Then our image is look like as in Figure 2.

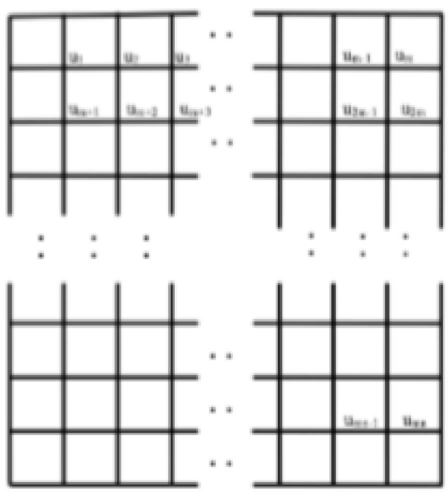

Figure 2. General 2D grid with n rows and m column

Now consider the general matrix for A for size of inpainting domain.

$$A = \begin{bmatrix} B & I & O & \dots & O \\ I & B & I & \dots & O \\ O & I & B & \dots & O \\ O & \dots & \dots & \dots & O \\ & \dots & \dots & \dots & \dots \\ & \dots & \dots & \dots & \dots \\ & \dots & I & B & I \\ & \dots & \dots & \dots B & I \end{bmatrix}$$

Where A is a matrix with nm × nm. It has n number of block matrix in each row and column.

$$B = \begin{pmatrix} -4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ 0 & 1 & -4 & 1 & & \\ & & & \ddots & & \\ & & & 1 & -4 & 1 \\ & & & & 1 & -4 \end{pmatrix}$$

B is a m × m matrix. Where m is a number of columns in inside of the grid.

$$I = \begin{pmatrix} 1 & 0 & & & \\ 0 & 1 & 0 & & \\ & & \ddots & & \\ & & 0 & 1 & 0 \\ & & & 0 & 1 \end{pmatrix}$$

$$O = \begin{pmatrix} 0 & 0 & & & \\ 0 & 0 & 0 & & \\ & & \ddots & & \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \end{pmatrix}$$

When we apply this method to our image, we did some development. We did image in-painting technique level by level. That is;

B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4	U5	U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 3. 5-point apply level by level

We have the boundary in formation and we have to find u1, . . . , u9. Normal 5-point method we need only one adjacent boundary level data. But in this method, we need two adjacent boundary level data to fill inside data.

Step 1:

B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4		U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 4. 5-point apply to level 1

Using the boundary data (red data) we can calculate adjacent level data (green data). Here we apply 5-point stencil method in different way;

For U_2

$$U_2 = 4B_{11} - B_4 - B_{10} - B_{12}$$

For U_4

$$U_4 = 4B_{20} - B_{16} - B_{19} - B_{24}$$

For U_6

$$U_6 = 4B_{21} - B_{17} - B_{22} - B_{25}$$

For U_8

$$U_8 = 4B_{30} - B_{29} - B_{31} - B_{37}$$

Step 2:

We use two different formulas for corner points and consider the average value. Such as;

For U_1

$$U_1^1 = 4B_{16} - B_9 - B_{15} - B_{20}$$
$$U_1^2 = 4B_{10} - B_3 - B_9 - B_{11}$$
$$U_1 = \frac{U_1^1 + U_1^2}{2}$$

For U_3

$$U_3^1 = 4B_{12} - B_5 - B_{11} - B_{13}$$
$$U_3^2 = 4B_{17} - B_{13} - B_{18} - B_{21}$$
$$U_3 = \frac{U_3^1 + U_3^2}{2}$$

For U_7

$$U_7^1 = 4B_{24} - B_{20} - B_{23} - B_{28}$$
$$U_7^2 = 4B_{29} - B_{28} - B_{30} - B_{36}$$
$$U_7 = \frac{U_7^1 + U_7^2}{2}$$

For U_9

$$U_9^1 = 4B_{25} - B_{21} - B_{26} - B_{32}$$
$$U_9^2 = 4B_{31} - B_{30} - B_{32} - B_{38}$$
$$U_9 = \frac{U_9^1 + U_9^2}{2}$$

Step 3:

When we are finding next level of data (blue data), we use the adjacent level data (green data);

$$U_5 = \frac{U_2 + U_4 + U_6 + U_8}{4}$$

B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4	U5	U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 5. 5-point apply to level 2

Now we compare our method with the existing method.

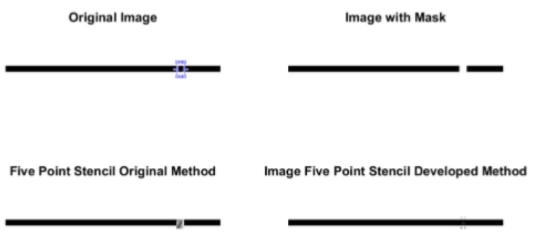

Figure 6. Standard 5-point method and modified 5-point method

When we check the two different approaches of 5 point stencil methods, we can see modified method PSNR value is larger than that of the standard 5-point stencil method. So, we use the modified 5-point stencil method values for our further calculations.

Table 1. PSNR value for standard 5-point method and modified 5-point method

	Standard 5-point method	Modified 5-point method
PSNR	29.1787	39.2958

see modified method PSNR value is larger than that of the standard 5-point stencil method. So, we use the modified 5-point stencil method values for our further calculations.

4. Linear iterative PDE method

In 2009, [2] C. K. Chui developed a multiresolution approximation method for image inpainting and surface completion. Here they use partial differential equation of anisotropic diffusion to know data.

$$\frac{\partial}{\partial t} u_j = \nabla \cdot (c(|\nabla u_{j-1}|) \nabla u_j) \ in\ D, t \geq 0$$
$$\frac{\partial}{\partial \mathbf{n}} u_j \Big|_{\partial d} = 0$$
$$u_j(z,0) = u_0(z), \qquad z \in D$$

Where j=1, 2, ... and $c(|\nabla u^j(z)|$ is the diffusion conductivity.
Here we have a set of linear partial differential equations. This is also called local image inpainting method.
Now we are going to solve this linear iterative equations.
when j=1;

$$\frac{\partial}{\partial t} u_1 = \nabla \cdot (c(|\nabla u_0|) \nabla u_1) \ in\ D, t \geq 0$$
$$\frac{\partial}{\partial \mathbf{n}} u_1 \Big|_{\partial d} = 0$$
$$u_1(z,0) = u_0(z), \qquad z \in D$$

Now apply the equation for this PDE,

$$u_1^{(k+1)} = u_1^{(k)} - \tau_k f(u_1^{(k)}) \quad k = 0, 1, 2, 3, ...$$

Now $f(u_1) = \nabla \cdot (c(|\nabla u_0|) \nabla u_1)$

Therefore,

$$u_1^{(k+1)} = u_1^{(k)} - \tau_k \nabla \cdot (c(|\nabla u_0|) \nabla u_1^{(k)})$$
$$k = 0, 1, 2, 3, ...$$

when k=0;

$$u_1^{(1)} = u_1^{(0)} - \tau_0 \bigtriangledown \cdot (c(| \bigtriangledown u_0 |) \bigtriangledown u_1^{(0)})$$
$$where \quad u_1^{(0)} = u_0$$

when k=1;

$$u_1^{(2)} = u_1^{(1)} - \tau_1 \bigtriangledown \cdot (c(| \bigtriangledown u_0 |) \bigtriangledown u_1^{(1)})$$

when k=2;

$$u_1^{(3)} = u_1^{(2)} - \tau_2 \bigtriangledown \cdot (c(| \bigtriangledown u_0 |) \bigtriangledown u_1^{(2)})$$
$$....$$

We can continue this process and we can get a value of u_1. We use this value to find u_2.

when j=2;

$$\frac{\partial}{\partial t} u_2 = \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2) \ in \ D, t \geq 0$$
$$\frac{\partial}{\partial \mathbf{n}} u_2 \Big|_{\partial d} = 0$$
$$u_2(z, 0) = u_0(z), \qquad z \in D$$

Now consider $f(u_2) = \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2)$
Therefore,

$$u_2^{(k+1)} = u_2^{(k)} - \tau_k \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2^{(k)})$$
$$k = 0, 1, 2, 3, ...$$

When k=0;

$$u_2^{(1)} = u_2^{(0)} - \tau_0 \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2^{(0)})$$
$$where \quad u_2^{(0)} = u_1$$

when k=1;

$$u_2^{(2)} = u_2^{(1)} - \tau_1 \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2^{(1)})$$

when k=2;

$$u_2^{(3)} = u_2^{(2)} - \tau_2 \bigtriangledown \cdot (c(| \bigtriangledown u_1 |) \bigtriangledown u_2^{(2)})$$
$$....$$

We can continue this process and we can get a value of u_2. We use this value to find u_3.

when j=3;

$$\frac{\partial}{\partial t} u_3 = \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3) \ in \ D, t \geq 0$$
$$\frac{\partial}{\partial \mathbf{n}} u_3 \Big|_{\partial d} = 0$$
$$u_3(z, 0) = u_0(z), \qquad z \in D$$

Now consider $f(u_3) = \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3)$
Therefor,

$$u_3^{(k+1)} = u_3^{(k)} - \tau_k \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3^{(k)})$$
$$k = 0, 1, 2, 3, ...$$

when k=0;

$$u_3^{(1)} = u_3^{(0)} - \tau_0 \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3^{(0)})$$
$$where \quad u_3^{(0)} = u_2$$

when k=1;

$$u_3^{(2)} = u_3^{(1)} - \tau_1 \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3^{(1)})$$

when k=2;

$$u_3^{(3)} = u_3^{(2)} - \tau_2 \bigtriangledown \cdot (c(| \bigtriangledown u_2 |) \bigtriangledown u_3^{(2)})$$
$$....$$

We can continue this process and we can get a value of u3. Using this method, we can solve linear iterative PDE.

Initial value of the non-linear and linear iterative PDE method is the 5-point stencil method values. Using a MATLAB program, we inpainted the damaged image. Here we compared MATLAB out puts for different in painting methods with different diffusion conductivity.

When we compare the inapinted image with the original image, we use PSNR values. Which is Peak Signal Noise Ratio. We define PSNR using mean squared error (MSE) and formula is given by,

$$PSNR = 20 \cdot log_{10} \frac{MAX}{\sqrt{MSE}}.$$

Where MAX is the maximum possible pixel value of the image.

Case 1: Linear iterative and nonlinear image inpainting PDE with constant diffusion conductivity

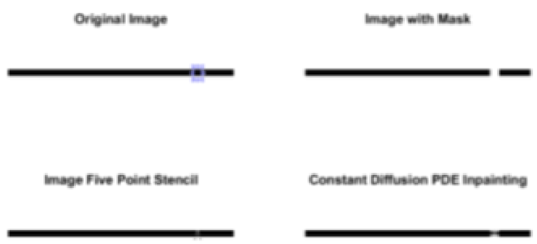

Figure 7. Image inpainting methods with constant diffusion

Table 2: PSNR value for linear iterative PDE and non-linear image inpainting PDE with constant diffusion conductivity

Inpainting methods	5-point stencil method	Constant conductivity PDE
PSNR	35.5747	29.9493

Here we consider the constant conductivity. That is,

c(p)= c. When c is a constant there is a no difference between linear iterative PDE and non-linear PDE.

That is, Linear iterative PDE,

$$\frac{\partial}{\partial t}u_i = \bigtriangledown \cdot (c) \bigtriangledown u_i) = c\Delta u_i$$

Non-linear PDE,

$$\frac{\partial}{\partial t}u = \bigtriangledown \cdot (c) \bigtriangledown u) = c\Delta u$$

When c is a constant, we have worst inpainted image. Therefore 5-point stencil method is better than the constant diffusion conductivity PDE.

Case 2: Linear iterative PDE and non-linear image inpainting PDE with inverse proportional diffusion conductivity

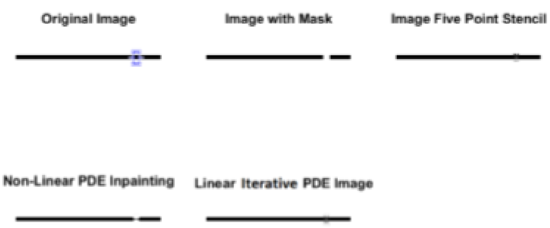

Figure 8. Image inpainting methods with inverse proportional diffusion conductivity

Here we consider the inverse proportional conductivity. That is,

c(p)= $\frac{1}{p}$.

With this diffusion conductivity,diffusion PDE is called TV inpainting method.

When we use this in MATLAB , we consider

c(p)= $\frac{1}{\varepsilon+p}$. Because we want to ignore the value of p=0.

Table 3: PSNR value for linear iterative PDE and non-linear image inpainting PDE with inverse proportional diffusion conductivity

Inpainting methods	5-point stencil method	Non-linear PDE	Linear iterative PDE
PSNR	30.3192	26.3699	31.9985

We use the 5-point stencil values as an initial value of Linear iterative PDE and nonlinear PDE. When we study this table, we can see PSNR value is increase to 5-point stencil method to linear iterative PDE method. Also, PSNR value of 5-point stencil method to linear iterative PDE method is decreasing. That is our inpainted image is worse than the 5point stencil method inpainted image. Therefore, linear iterative PDE method gave a better inpainted image.

Case 3: Linear iterative PDE and non-linear image inpainting PDE with Gaussian diffusion conductivity

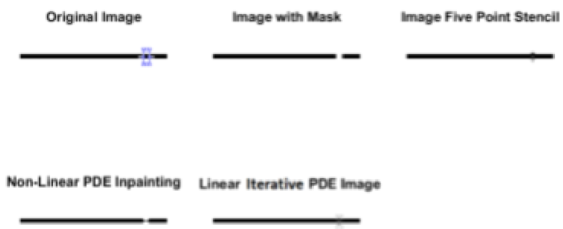

Figure 9. Image inpainting methods with Gaussian diffusion conductivity

Table 4. PSNR value for linear iterative PDE and non-linear image inpainting PDE with Gaussian proportional diffusion conductivity

Inpainting methods	5-point stencil method	Non-linear PDE	Linear iterative PDE
PSNR	29.9851	27.0343	33.1426

Here we consider the Gaussian Diffusion Conductivity. That is,

c(p)= $e^{-\frac{p^2}{k^2}}$.

Here also, we use the5-pointstencilvalues as an initial value of linear iterative PDE and non-linear PDE.

When we study this table, we can see PSNR value is increased to 5-point stencil method to linear iterative PDE method. But PSNR value of 5-point stencil method is decreased in the non-linear PDE method. That is non-linear method given worst inpainted image. Therefore, linear iterative PDE method gave the best inpainted image

Case 4: Linear iterative PDE and non-linear image inpainting PDE with Lorentz diffusion conductivity

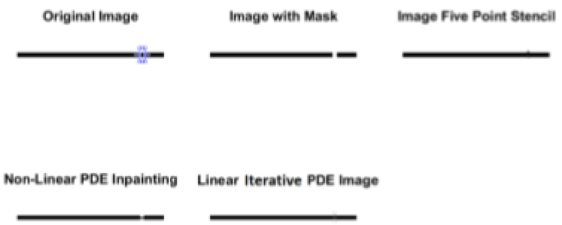

Figure 10. Image inpainting methods with Lorentz diffusion conductivity

Table 5. PSNR value for linear iterative PDE and non-linear image inpainting PDE with Lorentz proportional diffusion conductivity

Inpainting methods	5-point stencil method	Non-linear PDE	Linear iterative PDE
PSNR	34.7837	29.9902	37.3648

Here we consider the Lorentz Diffusion Conductivity. That is,

$$c(p) = \frac{1}{1+\frac{p^2}{k^2}}.$$

Here also, we use the 5-pointstencilvalues as a initial value of linear iterative PDE and non-linear PDE. When we study this table, we have same idea of previous methods. That is linear iterative PDE method gave the best inpainted image.

5. Conclusion

In this paper we solved Poisson's equation using 5-point stencil method. Here we used modified 5-point stencil method. It gave better results than the standard 5-point stencil method. Using modified 5-point stencil method values as an initial condition, we solve liner iterative and non-linear diffusion PDE. Also, here we consider different diffusion conductivity and compare their results. Using a PSNR value, we can see Linear iterative diffusion PDE method given the better results. Also, constant diffusion conductivity PDE had the worst results. Inverse diffusion conductivity PDE is better than that of the constant diffusion PDE. But it is worse than that of the Gaussian and Lorentz diffusion conductivity PDE. Gaussian and Lorentz diffusion conductivity Linear iterative PDE gave the better results for image inpainting.

6. References

[1] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C., (2000). Image inpainting. SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques, Wesley Publishing Co., NewYork, NY, USA, pages 417-424, 2000.

[2] Chui, C.K., (2009). An MRA approach to surface completion and image inpainting, Appllied and Computational Harmonic Analysis, Elsevier Science, Vol. 26, pages 270-276.

[3] Perona, P., Malik, J., (1990). Scale space and edge detection using anistropic diffusion, IEEE PAML, Vol. 12, pages 629-639.