



DYNAMIC IMAGE RESTORATION USING PARTIAL DIFFERENTIAL EQUATIONS

Avnish Kumar

Nitin Agrawal

Prashant Sinha

Semester III, B.Tech (Innovation with Mathematics & IT)

Under the supervision of Dr. Shobha Bagai

Cluster Innovation Centre

University Of Delhi



TABLE OF CONTENTS

Abstract	2
Acknowledgement	3
Introduction	4
Chapter 1: Laplace Equations.....	5
Chapter 2: Modeling of image color intensity dynamics using Heat and Laplace equations	7
Chapter 3: Methodology	10
Discretization of the Laplace's equation / Numerical method for Laplace equation	10
Chapter 4: Results	13
Future Scope	15
Appendix: Matlab Code.....	16
References	22

ABSTRACT

The current study aims at developing an inpainting system for restoration of lost art, reconstruction of destroyed images and removal unnecessary objects. The same has been achieved through Partial Differential Equations. Largely, the Laplace equation has been used clubbed with the Dirichlet boundary conditions in order to fill the inconsistencies/ destruction in the image using the colour intensities of the neighboring pixels as the boundary conditions. The results achieved were quite satisfactory in nature.

ACKNOWLEDGEMENT

First and foremost, we would like to thank Dr. Shobha Bagai, Associate Professor, Cluster Innovation Center for her guidance in dealing with this project. Her willingness to motivate us contributed tremendously to our project. We would also love to thank other faculty members at cluster Innovation Centre for their support. We are highly obliged by the opportunity that the university has provided us with. Last but not the least, we are thankful to our colleagues for their constant support for the fulfillment of this project.

INTRODUCTION

Partial differential equations are often used to model most of the basic theories underlying physics and engineering. We are enabled to simulate various phenomenon comprising a whole range of domains like electronics to image processing. We will be dealing with the Laplace equations in this paper as part of the method to achieve image inpainting.

Inpainting is the digital art of reconstructing old images by filling up the missing portions in a more legible and unified manner. The process uses various types of differential equations to delete an unwanted object in the image or fill up a break in the picture. Also referred as image interpolation, inpainting uses sophisticated algorithm to correct or replace lost or corrupted data in a image. The process has huge importance in restoring lost artefacts and paintings which have deteriorated over the course of time.

Using pde's as an algorithm to achieve the task of restoration is very prevalent these days and evolving very frequently. A numerical method discretizing the Laplace equations help us to replace corrupted pixels in the image matrix by running numerous iterations of the formulae. We present a very simple algorithm using Numerical methods for Laplace equations as a presentable software using graphic user interface.

CHAPTER 1: LAPLACE EQUATIONS

Laplace equation is a second order partial differential equation named after Pierre- Simon Laplace. It falls under the category of elliptic partial differential equations with the solutions known as harmonic functions. The general theory of solutions to Laplace's equation is known as potential theory. The solutions of Laplace's equation are the harmonic functions, which are important in many fields of science, notably the fields of electromagnetism, astronomy, and fluid dynamics, because they can be used to accurately describe the behavior of electric, gravitational, and fluid potentials. In the study of heat conduction, the Laplace equation is the steady-state heat equation.

Expression:

$$\Delta f = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} + \frac{d^2 f}{dz^2} = 0$$

In case of Laplace equations of two independent variables, the expression turns out to be

$$\frac{d^2 \varphi}{dx^2} + \frac{d^2 \varphi}{dy^2} = 0$$

Imposing Boundary Conditions:

The Dirichlet problem for Laplace's equation consists of finding a solution on some domain D such that φ on the boundary of D is equal to some given function. This means the function itself defined on the boundaries of the domain taken.

The Neumann boundary conditions for Laplace's equation specify not the function φ itself on the boundary of D , but its normal derivative. Physically, this corresponds to the construction of a potential for a vector field whose effect is known at the boundary of D alone.

Solutions:

The solutions of the Laplace equations are known as Harmonic functions and are analytic in nature. The solutions also follow the property of superposition, i.e. if two analytic functions in the valid domain of the equation are solutions to the Laplace equation, then their sum is also a solution of the same equation.

The easiest way to come up with an analytical solution of a 2D Laplace equation is by converting it into a set of ordinary differential equations. Let us assume the function φ to be product of independent functions of x and y.

$$\varphi(x, y) = X(x) * Y(y)$$

$$\text{So, } \frac{d^2\varphi}{dx^2} = Y(y) * \frac{d^2X}{dx^2} \quad \text{and} \quad \frac{d^2\varphi}{dy^2} = X(x) * \frac{d^2Y}{dy^2}$$

$$\Rightarrow \frac{d^2\varphi}{dx^2} + \frac{d^2\varphi}{dy^2} = \frac{1}{X} * \frac{d^2X}{dx^2} + \frac{1}{Y} * \frac{d^2Y}{dy^2} = 0$$

Since, X and Y are independent, we know that each term would be a constant. Thus our partial differential equation gets converted to a set of ordinary differential equations which can be easily solved to get the following general solution.

$$Y = \alpha \cos \lambda y + \beta \sin \lambda y \quad \text{and} \quad X = \alpha' \cos \lambda x + \beta' \sin \lambda x.$$

As these are analytic solutions, their sum is also a solution to the given Laplace equation.

CHAPTER 2: MODELING OF IMAGE COLOR INTENSITY DYNAMICS USING HEAT AND LAPLACE EQUATIONS

Inpainting is the technique of modifying an image in an undetectable way. It has a variety of applications from the restoration of damaged photos to the removal of selected objects in images. Moreover, it can also be applied to video applications such as fixing scratches or blots, removing logos or subtitles and error concealment. In theory, the inpainting problem is clearly ill-posed. The information inside the inpainting domain is totally lost and there is no unique solution. To solve the problem, different assumptions are declared to build the relation between the missing part and its known neighborhood.

Categorization of image inpainting algorithms

1. Structural Inpainting

Structural inpainting uses geometric approaches for filling in the missing information in the region which should be inpainted. These algorithms focus on the consistency of the geometric structure.

2. Textual Inpainting

Texture has a repetitive pattern which means that a missing portion cannot be restored by continuing the level lines into the gap. These algorithms come handy when structural inpainting algorithms are not able to restore image's texture due to its own limitations

3. Combined textual & structural Inpainting

In most advanced state of the art systems, a combination of both textual and structural inpainting is used to achieve desired results. Most parts of an image consist of texture and structure. The boundaries between image regions accumulate structural information which is a complex phenomenon. This is the result when blending different textures together.

We shall be using partial differential equations for the purpose of inpainting with Dirichlet boundary conditions.

Now, we need to look for a methodology to fill the gaps in an image for its perspective reconstruction. The heat equation fits the bill well. Fourier's heat equation describes how temperature in a material changes. Given this PDE, knowledge of the thermal conductivity of the material, and knowledge of the initial temperature conditions $u(t=0)$, it is possible to calculate the temperature at any point in the material in 2 – dimension for any given time by

$$u_t = c^2 \Delta^2 u = c^2 (u_{xx} + u_{yy})$$

Therefore heat equation could be a possible approach to this problem, wherein we shall be iterating the process for different time intervals. However, a more precise and intelligent approach would be considering a scenario wherein the time tends to infinity i.e. the temperature has been distributed well. This scenario could be better modeled using the steady state heat equation or the Laplace equation which is independent of time. The Laplace equation is given as follows

$$\Delta^2 u = u_{xx} + u_{yy} = 0$$

As discussed by Marcelo Bertalmio et al. in [1], we shall in fact be using his approach for inpainting through the Laplace equation i.e. the steady state heat equation. The approach has been modified at various places for improvement/ adaptability purposes.

The heat equation is also known as the diffusion equation owing to the very fact that besides modeling temperature dynamics, the equation could also present a model for describing density dynamics in a material undergoing diffusion. It can also be used to describe processes exhibiting diffusive-like behaviour, for instance the 'diffusion' of alleles in a population in population genetics. The Laplace equation thus presents a post diffusion scenario.

In case of image analysis the heat and the Laplace equation fits well. The intensity of each color in a RGB image or grayscale intensity in a b/w image could serve as the temperature or the material susceptible to diffusion for the purpose of filling up the holes in the image. Thus the color intensity of the area surrounding the hole / the portion which is to be removed / reconstructed is could be used to diffuse itself into the hole. Thus, the surrounding pixels could act as the boundary values i.e. they shall enable imposition of Dirichlet boundary conditions to the Laplace equation. The algorithms and the methodology used is described in detail in the following section

CHAPTER 3: METHODOLOGY

As discussed in the previous section we shall be using the Laplace equation with Dirichlet boundary conditions for the purpose of image inpainting. This section shall focus on discussing the methodology in detail.

Since images are represented as matrices wherein each element represents a pixel, they essentially are of a discrete nature. On the other hand partial differential equations or other such like modeling equation requires a continuous domain. Therefore in order to apply the Laplace equation to an image, we must either develop a continuous counterpart of the image or discretize the PDE model. Herein, we shall choose the later.

Discretization of the Laplace's equation / Numerical method for Laplace equation

For a PDE/ ODE in domain $x \in [a, b]$

$$\Delta x = (b - a)/N$$

Wherein for a function u . The derivatives of 'u' leads to N equation for u_i $i=1,2,3...N$

$$u_i \equiv u(i\Delta x)$$

$$x_i = i\Delta x$$

$$y_i = i\Delta y$$

Now, we shall approximate the second partial derivatives in the PDE using the central difference scheme,

$$\frac{d^2u}{dx^2}_{i,j} \approx \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2}$$

$$\frac{d^2u}{dy^2}_{i,j} \approx \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\Delta y)^2}$$

The above equations are synonymous to

$$\frac{d^2u}{dx^2} \approx \frac{u(x - \Delta x, y) - 2u(x, y) + u(x + \Delta x, y)}{(\Delta x)^2}$$

$$\frac{d^2u}{dy^2} \approx \frac{u(x, y - \Delta y) - 2u(x, y) + u(x, y + \Delta y)}{(\Delta y)^2}$$

Putting the double partial derivatives in the Laplace equation

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\Delta x)^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\Delta y)^2} = 0 \text{ for pixel } (i, j)$$

For $\Delta x = \Delta y$

The equation can further be rearranged as follows

$$-4u_{i,j} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} = 0 \text{ at pixel } (i, j)$$

Thus, we arrive at the discretized form of the Laplace equation as above.

Now, in an image with an inconsistency we shall essentially manually/ automatically objectify a rectangular region around the inconsistency. This particular region shall serve as the hole which is supposed to be filled by the diffused surrounding pixels. We shall have the accurate color intensities of the surrounding pixels which shall serve as the Dirichlet boundary conditions.

Using the boundary conditions, we shall proceed with the discretized Laplace equation for processing of the subsequent pixels.

From the equation, derived so far we can notice that intensity of each pixel (i,j) in the hole region shall be governed by the intensity of pixels on the immediate left, right, top & bottom.

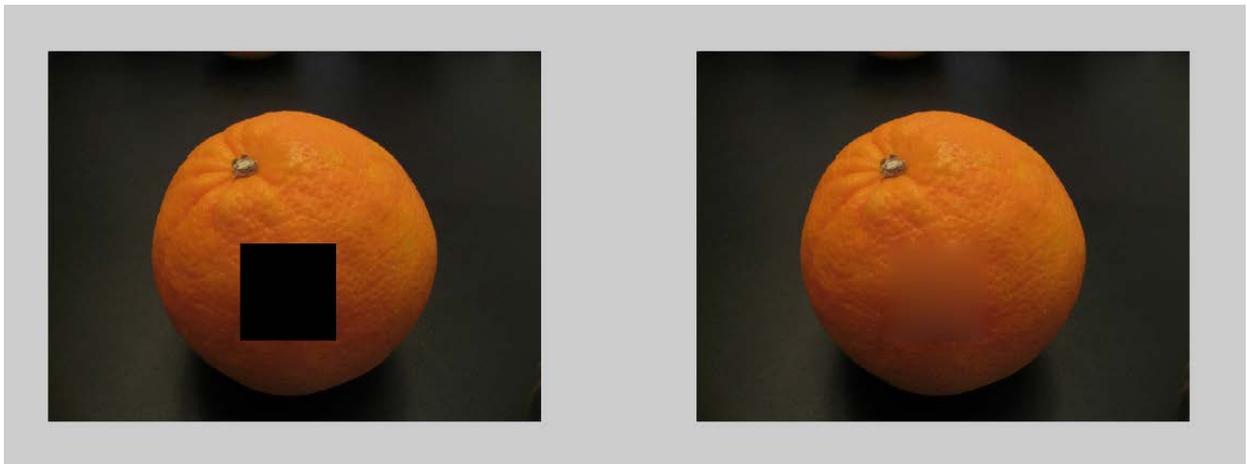
The process shall result in formation of multiple linear equations which can be further solved using Gaussian approach.

The process shall be repeated numerous times and on each of the pixels in order to achieve satisfactory results. The threshold for stopping could be calculated analytically or on the basis of the observations. In this way the requisite region of the image shall be inpainted.

CHAPTER 4: RESULTS

The results obtained were as follows.

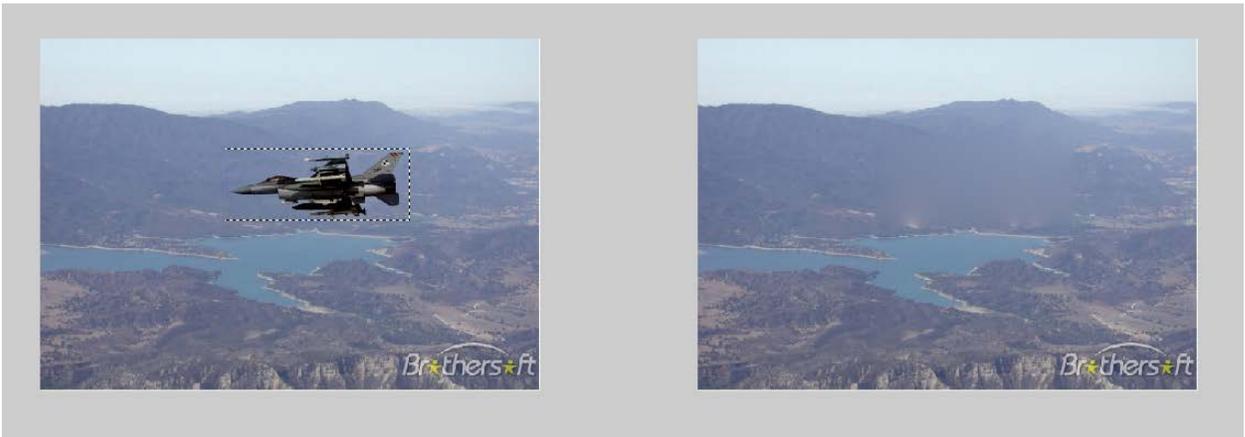
The image on the left shows the images which is to be inpainted while image on the right has been inpainted using our software. Under each of the images, the regions of their corresponding inconsistencies/ holes have been mentioned as was indicated to the code.



X = 393 393 608 608 ; Y = 601 393 393 601



X = 143 143 251 251 ; Y = 231 143 143 231



X = 220 220 467 467 ; Y = 230 121 121 240

As could be observed from the test cases above, the results were satisfactory and the software program developed using the discretized Laplace equation along with the Dirichlet boundary conditions could successfully undertake the requisite job.

FUTURE SCOPE

The model could further be extended to a combined structural and textural inpainting model for greater accuracy. The system could be critical in developing technologies and systems for reconstruction of lost art which could then be used for further analysis. The projects also presents an overview to how PDE's could be applied to vivid areas and hence open new arenas for confluence of Partial differential equations and Image processing

APPENDIX: MATLAB CODE

```
%%Deciding b/w RGB & grayscale images

function imgm = immain(I,xi,yi)

if(size(I,3)==3)

    imgm(:,:,1)=inpain3(I(:,:,1),xi,yi);

    imgm(:,:,2)=inpain3(I(:,:,2),xi,yi);

    imgm(:,:,3)=inpain3(I(:,:,3),xi,yi);

else

    imgm=inpain3(I,xi,yi);

end

end

%%Inpaining function

function img = inpain3(I,xi,yi)

final = I;

[r,c] = size(I);

mask = roipoly(r, c, xi, yi);

peri = bwperim(mask);

andar = mask & ~peri;

idx = find(andar);
```

```

if ~isempty(idx)

    blkgrd = zeros(size(mask));

    blkgrd(idx) = 1:length(idx);

    [m,n] = size(blkgrd);

    periv = zeros(m,n);

    periIdx = find(peri);

    periv(periIdx) = I(periIdx);

    right = zeros(m,n);

    right(2:(m-1),2:(n-1)) = periv(1:(m-2),2:(n-1)) + periv(3:m,2:(n-1)) + periv(2:(m-1),1:(n-2)) + periv(2:(m-1),3:n);

    right = right(idx);

    i = blkgrd(idx);

    j = blkgrd(idx);

    s = 4*ones(size(idx));

    for k = [-1 m 1 -m]

        Q = blkgrd(idx+k);

        q = find(Q);

        i = [i; blkgrd(idx(q))];

        j = [j; Q(q)];

        s = [s; -ones(length(q),1)];

    D = sparse(i,j,s);

```

```

end

D = sparse(i,j,s);

x = D \ right;

final(idx) = x;

end

img = final;

end

function varargout = Gui(varargin)

gui_Singleton = 1;

gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Gui_OpeningFcn, ...
                  'gui_OutputFcn',  @Gui_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

```

```
else

    gui_mainfcn(gui_State, varargin{:});

end

function Gui_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

axes(handles.axes1);

imshow('eight.tif');

function varargout = Gui_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```

```
function pushbutton1_Callback(hObject, eventdata, handles)

temp = uigetfile('*..*','Select the image file');

I=imread(temp);

axes(handles.axes1);

imshow(I);

rect=getrect;

floor(rect)

xmin=rect(1);

ymin=rect(2);

w=rect(3);

h=rect(4);

r=[xmin+h xmin xmin+h];

c=[ymin ymin ymin+w ymin+w];

axes(handles.axes2);

j=inpain3(I,r,c);

imshow(j)

function axes1_CreateFcn(hObject, eventdata, handles)
```

```
function axes2_CreateFcn(hObject, eventdata, handles)
```

REFERENCES

- <http://math.univ-lyon1.fr/~masnou/fichiers/publications/survey.pdf>
- <http://www.math.cornell.edu/~erin/docs/laplace.pdf>
- <http://www.tecn.upf.es/~mbertalmio/bertalmi.pdf>
- <http://www.slideshare.net/PulkitGoyal1/image-inpainting>
- An Image Inpainting Approach Based on the Poisson Equation ; Xiaowei Shao, Zhengkai Liu, Houqiang Li; Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)
- A. Criminisi, P. Perez and K. Toyama, "Object removal by exemplar-based inpainting", in Proc. IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 721-728, 2003.

