American University of Armenia

College of Science and Engineering

Numerical Analysis Project

PDE Methods in Image Processing

Harutyun Grigoryan
Liana Mehrabyan
Nataly Yayloyan

Fall, 2016

Abstract

This paper presents the application of partial differential equations in image processing, in particular, image restoration. The model that has been developed to process and restore partially damaged or scratched gray scale images, involves the numerical solution of heat equation using the method of finite differences. Application of the model on various images is discussed in addition to theoretical background and numerical solution.

Keywords: Image processing, image restoration, PDE, heat equation, numerical analysis, finite difference approximation, digital image.

Contents

Chapter 1

Introduction

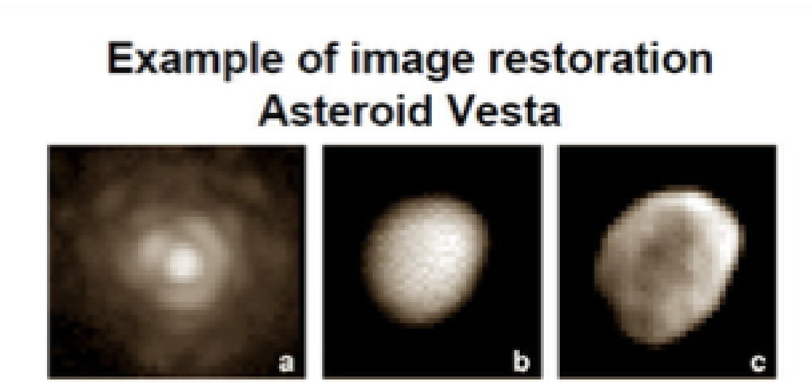
Image restoration is the process of recovering the original image from a given degraded or corrupt image. Those degradations usually include image noise, mis-focus, motion blur, partial damages, scratches etc. The latter is the problem that this project has elaborated. There are many methods developed during past decades for maximizing the efficiency of this process. Those methods include stochastic Modelization, Wavelets, Fourier Transforms etc. However, this paper will focus on one of the approaches instead, which includes the application of partial differential equations, in particular, numerical solution of heat equation.

1.1 What is Digital Image?

In order to be able to understand the rigorous statement of the problem, one needs to understand the concept of digital image and its pixel-based structure. A digital image is the numeric representation of an original image, consisting of small picture elements called pixels. Pixels are considered to be the smallest addressable elements of a digital image and their numeric values are stored in an ordered rectangular array. Dimensions of that array define the size of the digital image where the number of columns defines the image width and the number of rows defines its height accordingly. As a result, a digital image is perceived as a matrix having N columns and M rows. And, in order to address a certain pixel, one needs to address it by its i^{th} column and j^{th} row.($i \leq N$, $j \leq M$) While height and width define dimensions of an image, a third parameter is needed to fully define a digital image. Pixel intensity is used to describe the color or brightness of each pixel. This is the parameter that truly defines the image and makes it visible. Intensity value is defined by bits. In standard image digitalization, 8-bit range of intensity values if being used, i.e. intensity values vary from 0 to $2^8 - 1 = 255$. It is trivial to note, that if all pixels have the same intensity, an image will have a uniform shade of one color. In particular, a grayscale (also known as graylevel) image has pixels only with intensity values of all possible shades of grey, including black and white.

1.2 History of the Problem

Image restoration has considerable history and first attempts of image restoration date back to 1950s. First applications of image are noted in science and technology, in particular, in astronomy. Due to technical difficulties, first images of Moon and Mars were being delivered in large amount of degradations. And the necessity of retrieving as much information as possible gave a launch to adopting different algorithms and techniques of image restoration, which was later to become one of the most advanced fields of digital image processing.



Since then, many methods have been developed for solving the problem of restoration. Those methods are derived from the wide scope of mathematics, including Bayesian Methods from Probability and Statistics, method of Fourier transforms from Fourier analysis, Stochastic modelization based on Markov Random Field Theory. We suggest the interested reader to refer sources [1] and [2] for more thorough elaboration of the topic.

In the past two decades, one of the most studied and developed tools in image restoration is considered to be the PDE method. The main reason for that is the well-established and studied theory of PDEs. The fundamental PDE used in image restoration is the second order heat equation of the form:

$$\frac{\partial u}{\partial t}(t,z) = c\Delta u(t,z) \; t \geq 0, z \in R^2 \quad (1.2) \; *$$

With Dirichlet boundary condition:[1]

$$u(0,z) = u_0(z)$$

Where Δ is the Laplace operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, $u_0(z)$ is the initial degraded image, which is then transfigured by a process described by the equation (1.2). In addition, c is considered to be the diffusion coefficient. At this point, finding a function c that will give the best result is considered to be the crucial problem of this method and is often discussed in relevant scientific literature. One of the most advanced and often quoted choice of the function c is the proposal of Perona and Malik: c:[0,∞) → [0,∞) is a dicreasing smooth function function satisfying $c(0) = 1$. Other authors tried to elaborate the suggestion and regularize Perona and Malik model.[3]

1.3 The Structure of the Project Paper

In chapter 2, theoretical background of the problem will be discussed, i.e. the characteristics of heat equation and all the additional information required for digging into the numerical solution of the problem. Chapter 3 will be assigned to introduce the numerical solution of heat equation using the method of finite differences. The analogy of the image and spatial grid of heat equation will be presented, which will clarify the suitability of the solution method for image restoration. In chapter 4, practical application of the solution using the software MATLAB will be presented along with the discussion of the results on certain degraded photographs. The final chapter will conclude the project process and results and will mention possible elaboration that was not included in the scope of this paper and can be done in future research.

[1] A Direchlet boundary condition sets the variable we want to solve for to a specific value on that boundary.

Chapter 2

Theoretical Background

2.1 Heat Equation and its Analogies in Image Restoration

At its original physical meaning, heat equation governs the temperature distribution in an object. In particular, the 2-dimensional heat equation given by the form:

$$\frac{\partial u}{\partial t}(t,z) = k\Delta u(t,z) \quad (2.1)$$

where the constant k is called thermal diffusivity. The equation above is describing the temperature distribution on both x and y dimensions of a 2 dimensional object. An analogy of the diffusion coefficien c in equation (1.2) is easy to note. If the heat is initially concentrated in one place, equation (2.1) describes how the heat energy spreads out, which is a well-known phys-ical process called diffusion. That is why, the equation is often called diffusion equation. Initial and boundary conditions are often given along with the heat equation. Given Dirichlet boundary conditions describe the temperature along the edges of the object, whereas the initial condition $u(x,y,0) = f(x,y)$ is the initial state of the object at time 0. In the next chapter, the general numerical solution of the heat equation will be presented. Having already mentioned analogies in mind, one can apply this solution to the problem of image restoration.[5]

Chapter 3

Numerical Solution

3.1 Numerical Background

While solving differential equations, one can come up with infinitely many solutions. And in order to avoid that problem, equation is usually followed by initial and boundary conditions. Now, consider the following equation: "Heat Equation"

$$\frac{\partial u}{\partial t}(t,z) \text{ - } \Delta u(t,z) = 0 \text{ (3.1.1)}$$
$$u(0,z) = u_0(z)$$

where z at its turn has arguments x and y. The equation above is similar to the following one: [4]

$$u_t = c(u_{xx} + u_{yy}), 0 \leq x, y \leq 1, t \geq 0 \quad (3.1.2)$$

where $u_x x$ and $u_y y$ are double derivatives of u in terms of x and y, respectively, and c is the diffusion coefficient.

As it can be observed, $x, y \in [0,1]$. However, by symmetry the interval $[0,1]^2$ can be expanded to $[-1,1]^2$, in addition, by periodicity, to R^2. In given equations, $u(t,z)$ is the function for which the equation is being solved, where the argument t stands for time. $u_0(x)$ is initial condition. (it can be noted that $u(0,x)$ which is equal to $u_0(x)$ is solution of equation at time $t = 0$). The function $u(t_k, x_i, y_j)$, corresponds to the state of the grid $u(x_i, y_j)$ at time t_k. In order to start solving the equation for given function $u(t,z)$, spatial mesh points should be

$$(x_i, y_j) = (i * \Delta s, j * \Delta s), \text{ for } i, j = 0, 1, 2, ..., n-1 \text{ [4]}$$

where $\Delta s = \frac{1}{n-1}$. Now, for temporal mesh points:

$$t_k = k * \Delta t, \text{ for } k = 1, 2, \ldots \text{ [4]}$$

where Δt is being suitably chosen by user. Now, in order to approximate the solution of the given heat equation, following approximations should be done:

u_t: the derivative of u with respect to t
u_{xx}, the double derivative of u with respect to x
u_{yy}, the double derivative of u with respect to y

Calculations above should be done using the Taylor series expansion: (consider following equations for 1D)

$$\phi(x_i + \Delta x) = \phi(x_i) + \Delta x \left.\frac{\partial \phi}{\partial x}\right|_{x_i} + \frac{\Delta x^2}{2} \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{x_i} + \frac{\Delta x^3}{3!} \left.\frac{\partial^3 \phi}{\partial x^3}\right|_{x_i} + \cdots$$

One can get this when putting values of Δx to δx (considering their equality) Having, that $\Phi_i = \Phi(x_i)$ and $\Phi_{i+1} = \Phi(x_i + \Delta x)$ and using Mean value theorem(in order to replace higher order derivatives) equation can be written in following way

$$\frac{\Delta x^2}{2} \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{x_i} + \frac{(\Delta x)^3}{3!} \left.\frac{\partial^3 \phi}{\partial x^3}\right|_{x_i} + \cdots = \frac{\Delta x^2}{2} \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{\xi}$$

where $x_i \leq \xi \leq x_{i+1}$
Thus, for the final step equation is

$$\left.\frac{\partial \phi}{\partial x}\right|_{x_i} \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} + \frac{\Delta x^2}{2} \left.\frac{\partial^2 \phi}{\partial x^2}\right|_{\xi}$$

Similar steps can be done for 2D equation

We can substitute the corresponding derivatives of the function for any point (t_k, x_i, y_j) by (3.1.3)

$$u_t \approx \frac{{u_{i,j}}^{k+1} - {u_{i,j}}^{k}}{\Delta t} \text{ (3.1)}$$

$$u_{xx} \approx \frac{u^k_{i+1,j} - 2{u_{i,j}}^k + u^k_{i-1,j}}{(\Delta s)^2} \text{ (3.2)}$$

$$u_{yy} \approx \frac{u^k_{i,j+1} - 2{u_{i,j}}^k + u^k_{i,j-1}}{(\Delta s)^2} \text{ (3.3)}$$

Having (3.1.2) heat equation we plug approximations (3.1), (3.2), (3.3) into (3.3.2):

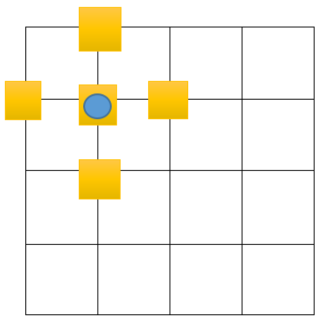
$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} = c\left(\frac{u_{i+1,j}^{k} - 2u_{i,j}^{k} + u_{i-1,j}^{k}}{(\Delta s)^2} + \frac{u_{i,j+1}^{k} - 2u_{i,j}^{k} + u_{i,j-1}^{k}}{(\Delta s)^2}\right)$$

So, for the last step, u_{ij}^{k+1} can be expressed as:

$$u_{i,j}^{k+1} = u_{i,j}^{k} + c\frac{\Delta t}{(\Delta s)^2}(u_{i+1,j}^{k} + u_{i-1,j}^{k} - 4u_{ij}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k}) \; *$$

3.2 Analogies of Heat Equation and Image Restoration

In Image Restoration, the equations (3.1.1) and (3.1.2) have the following meanings: $u_0(x)$ is the initial picture, selected by the user, $u(t,z)$ (z having arguments x,y) is the image after some restorations(after time t). Argument of u: z has following meaning in image restoration: it shows the place of each pixel of a picture, which allows us to understand that z itself has its arguments x and y (in order to show the position of each pixel). Coordinates (x_i, y_j) (for $i, j = 0, 1, 2, ., n-1$) show the positions of pixels of image, k and $k+1$ show the moments, k-exact, $k+1$-next moments($u_(i,j)^k$,$u_(i,j)^(k+1)$ is are showing the image before one restoration and same image after one restoration, respectively). Brief visualization of solution used on a matrix (image pixels)



Getting blue point for picture in step $k+1$ having pixels if an image in step k(yellow ones) n x n martix

Consider the spatial mesh $\Delta s = 1/n - 1$. * By this method, picture is being divided into equal squares and image restoration is being done on that squares.

Also, to solving equation numerically and to use solution in order to get final image, few more data are needed to be separated:

$u(0,0,y)$- 1^{st} column of pixels of a picture
$u(0,x,0)$- 1^{st} row of pixels of a picture
$u(0,n,y)$- last column of pixels of a picture
$u(0,x,n)$- last row of pixels of a picture

Consider the spatial mesh $\Delta s = \frac{1}{n-1}$. * By this method, picture is being divided into equal squares and image restoration is being done on that squares. As observed in the equation, $u_{i,j}$ is constructed using $u_{i-1,j}, u_{i+1,j}, u_{i,j-1}, u_{i,j+1}$(using pixels from previous row, previous column, next row and next column). However, this method will not work for the edges of the image as their previous or next rows or columns do not exist.

3.3 Solution for Rectangular matrices

Cases mentioned above in chapter 3 are all for square matrices. The following subsection will consider the case of rectangular matrices. Almost all calculations remain unchanged, only one change is being done in the idea of calculation: Consider $\Delta s = \frac{1}{n-1}$. The rectangular matrix will be divided not into squares but into equal rectangles and restoration is being done on that rectangles.

Chapter 4

Process and Results

4.1 The Process and Results

After deriving the formula by which the restoration of damaged pixels should be done, a program has been written using MATLAB software, in order to visualize the results. The script of the code can be found in the Appendix of the paper. The created function takes as its arguments the image that should be restored and the diffusion coefficient (i.e. c). Every image needs its diffusion coefficient, c, in order to maximize the result of restoration. That is why, the input of c is left to user preference. A detailed explanation of the written code will follow this section. First of all, the dimensions of the image are assigned to the variables m and n and are set to be the dimensions of the resulted image u_{new}. Now, the coefficient $c\ \frac{\Delta t}{\Delta s^2}$ should be constructed that will later be used in the formula of obtaining the shade of the new pixel in the resulted image.

Temporal mesh Δt should be suitable for every image. In case if Δt takes small values, it has little effects on the obtained image u_{new} even for some values the impact is not distinguishable, and for bigger values of Δt, the grey color exceeds the maximum intensity value 255 in every pixel and the image becomes just a grey image. After several trials, an optimal value of Δt was chosen which is 0.03. Spatial mesh Δs is taken to be $\frac{1}{n-1}$ as elaborated in the 3^{rd} chapter. Next, the variable t_{max} specifies how many times the loop of restoration of u_{new} should traverse the image array, and is equal to $2\Delta t$. (Implying that the loop will traverse twice)

Then, the boundary pixels of the initial image u_0 are assigned as the boundary parts of u_{new}, as elaborated in the 3^{rd} chapter. Finally, the inner pixels of the resulted image are obtained according to the formula obtained by solving the heat equation numerically. After filling the inner part of the image u_{new}, it is assigned to be u_0 in order to obtain the values of the damaged pixels precisely and it continues

until the stopping condition of the loop. Finally, the function returns the restored image.
Several images have been processed by this codes and the following results were obtained. In particular, zoomed sections of scratched parts are presented for a better visualisation.

Figure 4.1: Image 1

Figure 4.2: Image 2

Figure 4.3: Image 3

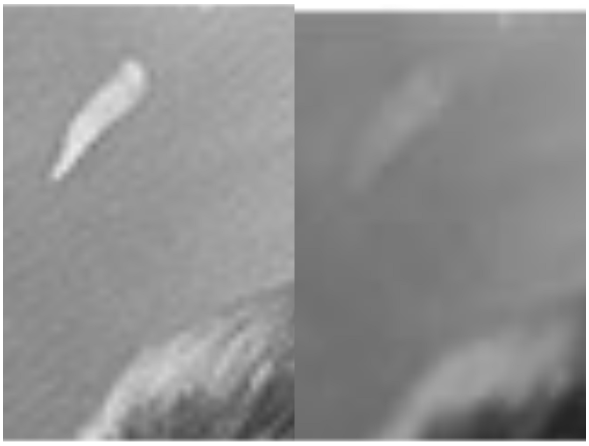


Figure 4.4: Image 1 pixels

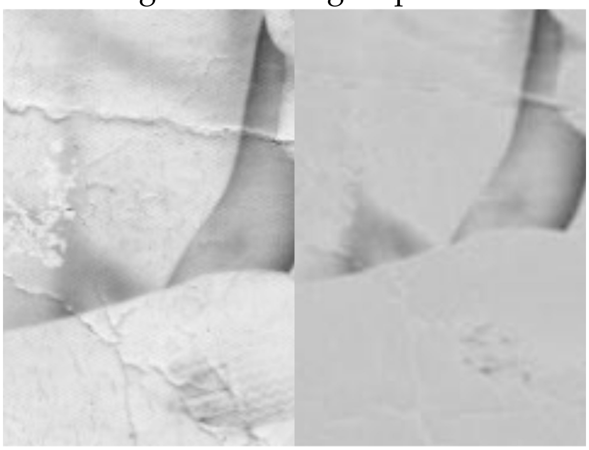


Figure 4.5: Image 2 pixels

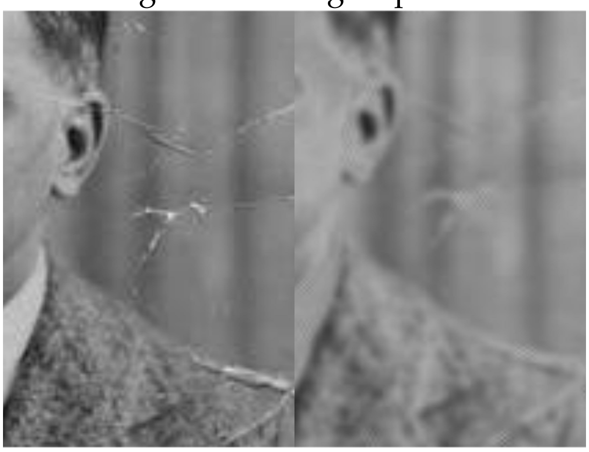


Figure 4.6: Image 3 pixels

Chapter 5

Conclusion

The aim of this project was to study and apply PDE methods in image restoration. Many methods have been developed to improve the results of image restoration which were derived from different fields of mathematics. And the most studied and fundamental method is the transformation of the initial degraded image by the process governed by the heat equation. From variety of image degradations, this project has focused on the restoration of partially damaged or scratched grayscale images. The team of this project has embarked for this work by doing prior research of the topic and understanding the theoretical knowledge and background necessary for applying the method. Several decade history of image processing has been briefly presented along with theoretical background and advanced methods of today, including the Perona Malik model. After learning necessary theory of Partial Differential Equations and their numeric approximation, a practical application has been applied to the theory using MATLAB software. Different degraded photographs have been processed by the program, the results of which have been presented in the 4th chapter of this paper. The importance of image restoration is crucial both in science and art. That is why it is remaining one of the most studied fields and many advancements and improvements can be done that were not covered in this paper.

5.1 Further Research and Suggestions

One of the findings that could significantly improve the results of this model would be the optimal diffusion coefficient c(s) already mentioned in Chapter 1. The model presented by Perona and Malik suggests to take a decreasing smooth function such that the equation will remain parabolic. However, the precise form of the function $c(s)$ still remains unfound. Regularization of Perona Malik model has also been proposed by M. Strboja. The idea of regularization stands in substituting the diffusion coefficient by a smooth version of it. $G_\sigma * \Delta u$ is a Gaussian smoothing kernel which would change the initial model to a model with the following form:

$$\frac{\partial u}{\partial t}(t,z) = div(c|\nabla G_\sigma * u|^2)\nabla u(t,z)$$
$$u(0,z) = u_0(z) *$$

Bibliography

[1] Molina R., Nunez J., "Image Restoration in Astronomy"
Retrieved from:
http://diposit.ub.edu/dspace/bitstream/2445/8546/1/500851.pdf
[2] Aubert G., Kornprobst P., "Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations"
Retrieved from:
https://caniban.files.wordpress.com/2011/04/mathematical-problems-in-image-processing.pdf
[3] Strboja M., "Image Restoration by Partial Differential Equations"
Retrieved from:
http://conf.uni-obuda.hu/sisy2006/25$_{M}$*irjana.pdf*
[4]*HorakV., GruberP., "ParralelNumericalSolutionof2DHeatEquation"*
Retrievedfrom :
https : //www.cosy.sbg.ac.at/events/parnum05/book/horak1.pdf
[5]*HabermanR., "AppliedPartialDifferentialEquationswithFourierSeriesandBoundary ValueProblems"*

Chapter 6

Appendix

```
PDE.m    PDEinput.m
function u_r = PDE(image, c)
u0=image;

[n,m,k]=size(u0);

delta_t=0.03;
delta_s=1/(n-1);
p=c*delta_t/delta_s^2;
tmax=2*delta_t;
t=0;
u_new=zeros(n,m);

for k=1:1:n %filling the boundaries for n
  u_new(k,1)=u0(k,1);
  u_new(k,m)=u0(k,m);
end

for k=1:1:m %filling the boundaries for m
  u_new(1,k)=u0(1,k);
  u_new(n,k)=u0(n,k);
end

while t<tmax
  for i = 2:1:n-1;

    for j = 2:1:m-1;
      u_new(i, j) = u0(i,j)+ p*u0(i+1, j) + p*u0(i-1, j) - 4*p*u0(i, j) + p*u0(i, j+1) + p*u0(i, j-1);
    end
  end
  u0 = u_new;
  t=t+delta_t;
end
u_r=u_new;
end
```

Figure 6.1: Matlab Function

```
PDE.m    PDEinput.m
u0=imread('Image1.jpg');
imshow(u0,[])
figure()
%Optimal values for c in chosen cases
%image 1 c=0.0001,
%image 2 c=0.00002,
%image 2 c=0.00008,
u_new=PDE(u0, 0.0001);
imshow(u_new,[])
```


Figure 6.2: Matlab Input