Journal homepage http://revistas.unitru.edu.pe/index.php/SSMM

SELECCIONES MATEMÁTICAS
Universidad Nacional de Trujillo
ISSN: 2411-1783 (Online)
2021; Vol. 8(2): 411-416.

MATHEMATICS EDUCATION

Finite difference methods in image processing

Esquemas de diferencias finitas en el procesamiento de imágenes

Juan Triana[*] and Luis Ferro[†]

Received, Oct. 01, 2021 *Accepted, Dec. 10, 2021*

How to cite this article:
Triana J, Ferro L. *Finite difference methods in image processing*. Selecciones Matemáticas. 2021;8(2):411–416.
http://dx.doi.org/10.17268/sel.mat.2021.02.17

Abstract

Digital Image processing has been a research area of interest in the last decades, standing out for its applications in the analysis of diagnostic images and astronomical images. In this paper, we perform an overview of edge detection methods through finite-difference to present edge detection as a problem-based learning strategy for numerical differentiation, in order to improve the students' skills in modeling and algorithmic thinking in numerical analysis courses. In addition, we present image restoration through finite-difference as a problem involving partial differential equations and software tools.

Keywords . Finite difference, edge detection, image restoration.

Resumen

El procesamiento de imágenes digitales ha sido un área de investigación de interés en las últimas décadas, destacándose por sus aplicaciones en el análisis de imágenes diagnósticas e imágenes astronómicas. En este artículo, realizamos una descripción general de los métodos de detección de bordes a través de diferencias finitas, con el fin de presentar la detección de bordes como una estrategia de enseñanza de los esquemas de diferencias finitas mediante aprendizaje basado en problemas, buscando desarrollar competencias de modelamiento matemático y pensamiento algorítmico en estudiantes de análisis numérico. Además, presentamos la restauración de imágenes mediante diferencias finitas como un problema que involucra ecuaciones diferenciales parciales y herramientas de software.

Palabras clave. Diferencias finitas, detección de bordes, restauración de imágenes.

Digital image processing consists of the manipulation of images using computational tools, involving several techniques over images such that enhancement, compression, edge detection, restoring, and many others. There are many applications of image processing in different areas such as biomedical and medical physics [6], biometric and signature verification [7], among others [11]. An image is usually conceived as a two-dimensional array of pixel values [9], therefore a matrix A can be used to represent a digital image, where $A_{i,j}$ represents the intensity of the pixel located in the position i, j of the image.

Visualization is considered as the graphical display of information to facilitate the learning of some knowledge. Learning outcomes in a numerical analysis course are related to using software tools and modeling applied problems; therefore is a course with several options for visualization. To improve computational thinking in the student, we have to choose interesting topics for solving applied problems through numerical methods. Thus we consider image processing using visualization as a didactic tool, establishing the connection between matrices and images.

Problem-based learning (PBL) can be used to improve numerical and analytical thinking ability in students [5]. It is necessary to consider a real-life problem to get the interest of the students, digital image

[*] Universidad ECCI, Bogotá, Colombia. (jtrianal@ecci.edu.co).
[†] Programa de Estadística, Universidad ECCI, Bogotá, Colombia. (lferroa@ecci.edu.co).

processing provides many interesting problems that can be considered through numerical analysis methods. Here we present some applications of partial differential equations to edge detection and image restoration through finite difference methods, we use Matlab to perform numerical methods but it can be replaced by Octave, the open-source alternative.

1. Mathematical background. Digital images can be represented through matrices; in the case of a grayscale image (for an 8-bit grayscale image), every pixel takes an entire value between 0 (black) and 255 (white) [12]. Considering the following random matrix

$$\begin{bmatrix} 252 & 89 & 32 & 118 & 206 & 235 & 81 & 154 & 57 & 207 \\ 43 & 175 & 6 & 54 & 191 & 203 & 208 & 99 & 95 & 98 \\ 66 & 75 & 74 & 25 & 31 & 147 & 201 & 36 & 22 & 157 \\ 101 & 135 & 81 & 210 & 134 & 112 & 217 & 6 & 163 & 147 \\ 19 & 212 & 167 & 45 & 83 & 66 & 129 & 107 & 46 & 135 \\ 174 & 152 & 244 & 42 & 139 & 192 & 162 & 47 & 11 & 70 \end{bmatrix},$$

we obtain the image representation

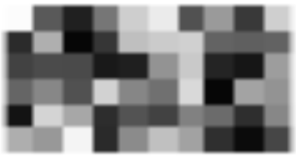

Figure 1.1: A grayscale image.

Figure 1.1 was obtained by Matlab with the instruction *imshow*. On the other hand, if we want to get the matrix representation of an image we need the instruction *imread*. The image capture is usually performed by devices that are sensitive to red, green, and blue parts of the light spectrum; therefore, the vast majority of images are considered as RGB images. In RGB color model, an image is represented with three grayscale images corresponding for each color component, i.e. a matrix for red intensity, a matrix for green intensity, and a matrix for blue intensity, these matrices are known as masks.

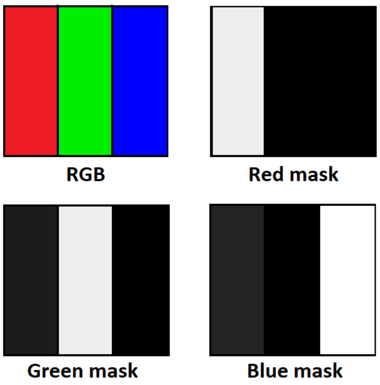

Figure 1.2: (a) RGB image. (b) Red mask. (c) Green mask. (d) Blue masks.

For instance, by applying the instruction *imread* in the form:

```
A=imread('imagen.png');
```

If image.png is a grayscale image A is a matrix; if image.png is a color image A is a 3D-array generated by 3 matrices, one matrix for each color mask.

As an interesting exercise, students may perform a transformation of an 3D-array to a matrix, to convert a color image to a grayscale image. In Figure 1.3, we use the instruction *rgb2gray*.

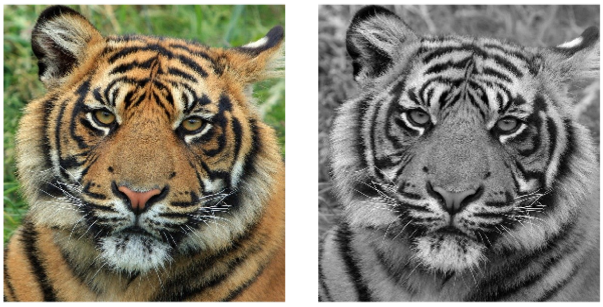

Figure 1.3: (a) Original image taken from pixabay. (b) grayscale image.

Some finite difference approximations of the first-order derivative of f are given by

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h), \tag{1.1}$$

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h), \tag{1.2}$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2). \tag{1.3}$$

The formulas above are known as forward difference, backward difference, and central difference, respectively. These schemes can be extended for several variables, e.g. for two variables, central difference schemes for the first derivative are given by

$$\frac{\partial f}{\partial x}(x,y) = \frac{f(x_{i+1},y_j) - f(x_{i-1},y_j)}{2h} + O(h^2), \tag{1.4}$$

$$\frac{\partial f}{\partial y}(x,y) = \frac{f(x_i,y_{j+1}) - f(x_i,y_{j-1})}{2k} + O(k^2). \tag{1.5}$$

In order to use index notation, we rewrite $f(x_i, y_j)$ as $f_{i,j}$.

2. Edge detection. Egde detection is a technique of image processing using to detect significant local changes in an image, it can be performed in color images [2], but the vast majority of edge detection filters are based on grayscale images; thus, for color images, we have to convert the image to grayscale cf. [16].

In image processing, the gradient vector is a measure of the change in pixel values; consequently, edge detection with first-order derivatives means to search in all directions for the largest value changes [8]. Let A be the matrix that represents the image; thus, the magnitude of the gradient in position i, j is given by

$$\sqrt{(A_{i+1,j} - A_{i,j})^2 + (A_{i,j+1} - A_{i,j})^2}. \tag{2.1}$$

Since $\sqrt{x^2 + y^2} \leq |x| + |y|$, we get an approximation for gradient vector that requires a lower computational cost

$$|A_{i+1,j} - A_{i,j}| + |A_{i,j+1} - A_{i,j}|. \tag{2.2}$$

Students can perform a method based on Equation (2.2), but sometimes the result is a black image; to solve it, we need to apply the transformation $T(A) = \alpha A$, with $\alpha > 1$, where A is the matrix representation of the edges. It is necessary to consider several values for α to find the best result. Student may deduce several methods similar to Equation (2.2) considering other finite difference schemes for the first derivative, and perform them in order to compare the obtained results.

The pseudo-Laplace operator is given by the discretization of $|\nabla f| = \left|\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right|$. Here we use central difference for second derivative,

$$\frac{\partial^2 f}{\partial x^2}(x,y) = \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2} + O(h^2), \tag{2.3}$$

$$\frac{\partial^2 f}{\partial y^2}(x,y) = \frac{f(x,y+k) - 2f(x,y) + f(x,y-k)}{k^2} + O(k^2). \tag{2.4}$$

In terms of index notation, considering $h = k = 1$, we get

$$\frac{\partial^2 f}{\partial x^2}(x_i,y_j) = f_{i+1,j} - 2f_{i,j} + f_{i-1,j} \quad \text{and} \quad \frac{\partial^2 f}{\partial y^2}(x_i,y_j) = f_{i,j+1} - 2f_{i,j} + f_{i,j-1}.$$

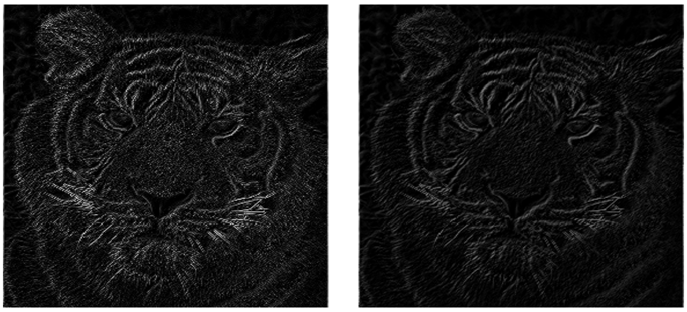

Figure 2.1: (a) Gradient method with forward differences. (b) Gradient method with central differences.

Therefore, if A is the matrix that represents the image, the pseudo-Laplace operator is given by

$$|A_{i+1,j} - A_{i-1,j} - 4A_{i,j} + A_{i,j+1} + A_{i,j-1}|\,.$$

It is called Laplace filter [8]. In terms of matrices we have the following representation, known as kernel, for the pseudo Laplace operator

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

There are many filters for edge detection, following [16] we may obtain the following filters and its respective kernels.

$$\text{Roberts: } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \text{Prewitt: } \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \qquad \text{Sobel: } \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}.$$

We may introduce 2D convolution to apply a filtering kernel on the image as an interesting exercise. Since Matlab provides us with the instruction *edge*, students can explore these filters directly.

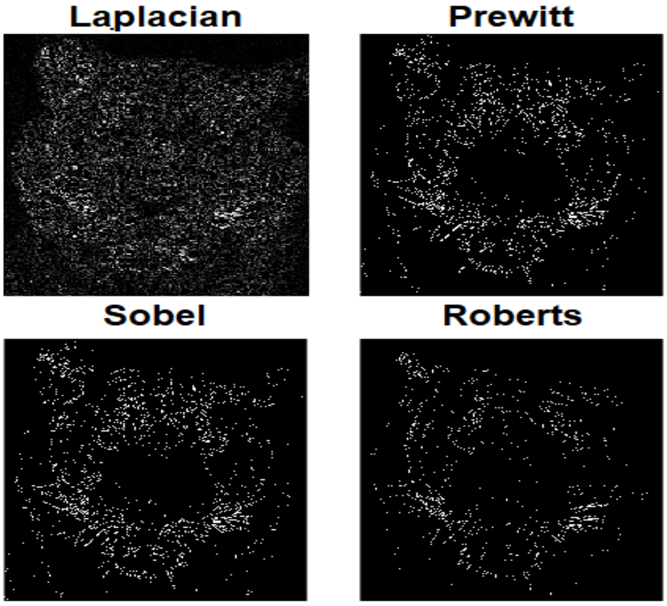

Figure 2.2: (a) Laplacian filter. (b) Prewitt filter. (c) Sobel filter. (d) Roberts filter.

Canny edge detection algorithm cf. [3], is one of the most popular methods in image processing. It can be applied by the instruction *edge* in the form

```
E = edge(A,'canny');
```

Where A is the matrix representation of the image, as result we obtain E as the matrix representation of the edges. Similar to Figure 2.2, students can perform a comparison among the methods of edge detection.

3. Image restoration. Image restoration can be defined as the process of recovering an image that has degraded, mathematical models for image restoration can be found in [13]. An interesting application of image processing is the art restoration; it can be used to guide the restoring process [14]. The restoration process consists of filling in the missing areas using information from the surrounding pixels; here we use a diffusion process by heat equation as a restoring method. The Equation (3.1) is known as heat equation

$$\frac{\partial A}{\partial t} = \lambda \left(\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} \right). \tag{3.1}$$

Where λ is a constant, x and y are spacial variables and t is the time [1]. Second order derivatives in the heat equation can be approximated by central difference and the time derivative can be approximated by forward difference, thus obtaining

$$\frac{A^{k+1}_{i,j} - A^k_{i,j}}{\Delta t} = \lambda \left(\frac{A^k_{i+1,j} - 2A^k_{i,j} + A^k_{i-1,j}}{\Delta x^2} + \frac{A^k_{i,j+1} - 2A^k_{i,j} + A^k_{i,j-1}}{\Delta y^2} \right).$$

Where A is the matrix representation of an image, $A^k_{i,j}$ is the value at time k, since it is a discrete time process we may consider $\Delta t = 1$. On the other hand, for images we get $\Delta x = \Delta y = 1$. Therefore, we get the explicit formula for heat equation

$$A^{k+1}_{ij} = \lambda \left(A^k_{i+1,j} + A^k_{i,j+1} - 4A^k_{i,j} + A^k_{i,j-1} + A^k_{i-1,j} \right) + A^k_{i,j}. \tag{3.2}$$

The iterative method given in Equation (3.2), is consistent with the partial differential equation cf. [15]. Numerical stability for this explicit method is given by $\Delta t \leq \frac{(\Delta x)^2 + (\Delta y)^2}{8\lambda}$ cf. [4]; thus, for $\lambda \leq \frac{1}{4}$ we get a stable method. Hence by Lax-Richtmyer equivalence theorem we get the convergence of the method [10].

Restoring process starts with the manual marking of the damaged regions, in order to use edge detection methods we suggest to perform a edge filter to detect automatically the marked regions. Once we have identified these regions, we apply the heat equation in the marked regions, here we use the explicit form presented in Equation (3.2).

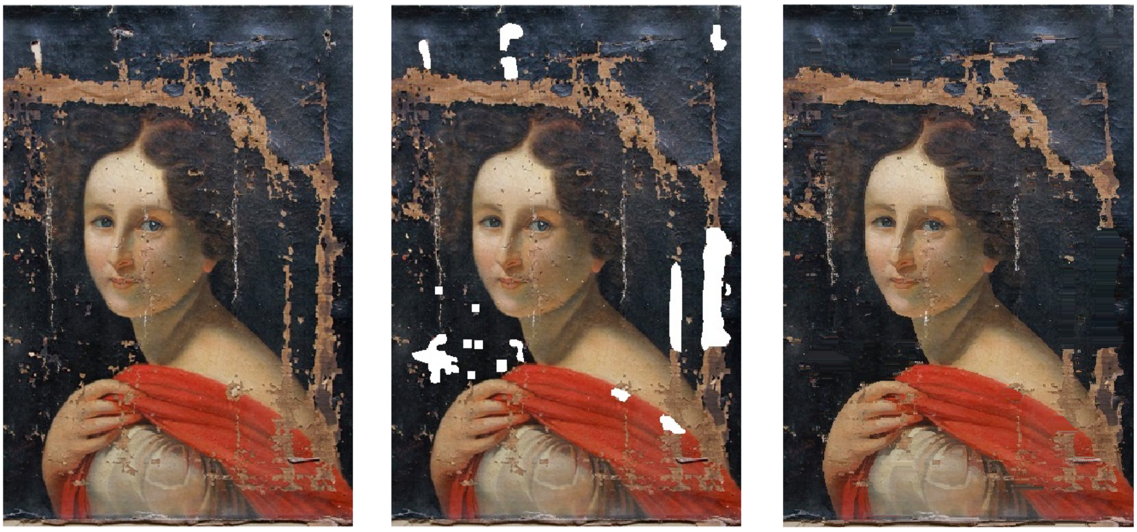

Figure 3.1: (a) Original image taken from https://www.amigosdar.org/taller-de-restauracion. (b) Some marked regions. (c) Restored image.

Students can explore considering several marked regions; furthermore, they can propose another method for apply heat equation, for instance an implicit method. In [17], detection of damaged regions in a image is performed through edge detection filters on the original image; however, it is a noise-sensitive method.

4. Conclusions. For a PBL activity considering edge detection, we suggest the following process: Select an image from internet. Perform the conversion to a grayscale image. In an applied problem, the students must select the appropriate method for solving the given problem and they have to implement the respective algorithm, here we suggest Laplacian filter because it is easier to understand for students. Finally, students may perform a comparison among several methods for edge detection via convolutions and kernels or by using the instruction *edge* of Matlab, this will depend on the competencies that teacher wants to develop in students.

A problem of image restoration requires skills with several instructions of image processing; in addition, the student has to deduce the algorithm of restoring, thus this activity could be considered very difficult. Therefore, we suggest this topic as a project, here the teacher's role will be to guide the student on the way to get the solution.

Finite-difference methods are essential for generating numerical methods applied to image processing, mainly due to the ease they offer in their programming with a low computational cost. The students can increase the level of understanding of finite differences through digital image processing, due to this area provides interesting problems for development of meaningful learning.

ORCID and License

Juan Triana https://orcid.org/0000-0003-2991-6082
Luis Ferro https://orcid.org/0000-0001-5678-4957

References

[1] Buchanan J, Shao Z. A first course in partial differential equations. Singapore: World Scientific, 2018.
[2] Burger W, Burge M. Principles of digital image processing. London: Springer, 2013.
[3] Canny J. A Computational Approach To Edge Detection. IEEE Trans. Pattern Anal. Mach. Intell. 1986; 8(6):679-698.
[4] Chapra S, Canale R. Numerical methods for engineers. 7th ed. New York: Mc Graw Hill, 2015.
[5] Darma I, Candiasa I, Sadia I, Dantes N. The effect of problem based learning model and authentic assessment on mathematical problem solving ability by using numeric ability as the covariable. J. Phys. Conf. Ser. 2018; 1040:1-9.
[6] Deserno T. Biomedical image processing. Berlin: Springer, 2011.
[7] Jabin S, Zareen F. Biometric signature verification. Int. J. biometrics. 2015; 7(2):97-118.
[8] Jahne B. Digital image processing. Berlin: Springer, 2002.
[9] Kinzer J. Image operators. Boca Raton: CRC Press, 2018.
[10] Li J, Chen Y. Computational partial differential equations using Matlab. 2nd ed. Boca Raton: CRC Press, 2020.
[11] Lukac R, Plataniotis K. Color image processing: methods and applications. Boca Raton: CRC Press, 2007.
[12] Petrou M, Petrou C. Image processing: The fundamental. 2nd ed. Singapore: John Wiley and Sons, 2010.
[13] Pratt W. Introduction to Digital image processing. Boca Raton: CRC Press, 2014.
[14] Schönlieb C. Partial differential equation methods for inpainting. New York: Cambridge University Press, 2015.
[15] Sewell G. The numerical solution of ordinary and partial differential equations. 2nd ed. New Jersey: John Wiley and Sons, 2005.
[16] Solomon C, Breckon T. Fundamentals of digital image processing. Oxford: Wiley-Blackwell, 2011.
[17] Triana J, Romero L, Roldan D. Restauración digital de imágenes mediante ecuaciones diferenciales parciales. Rev. U.D.C.A Act. & Div. Cient. 2013; 16(2):511-518.