Image Processing through Linear Algebra

Dany Joy
Department of Mathematics, College of Engineering Trivandrum
danyjoy4@gmail.com

ABSTRACT

The purpose of this project is to develop various advanced linear algebra techniques that apply to image processing. With the increasing use of computers and digital photography, being able to manipulate digital images efficiently and with greater freedom is extremely important. By applying the tools of linear algebra, we hope to improve the ability to process such images. We are especially interested in developing techniques that allow computers to manipulate images with the least amount of human guidance.

Keywords: Image processing, matrix addition, multiplication, substraction, rotation, edge detection, spatial domain techniques

1. INTRODUCTION

In this paper we develop the basic definitions and linear algebra concepts that lay the foundation for later chapters. And also demonstrate techniques that allow a computer to rotate an image to the correct orientation automatically, and similarly, for the computer to correct a certain class of colour distortion automatically. In both cases, we use certain properties of the eigenvalues and eigenvectors of covariance matrices. Also we model colour clashing and colour variation using a powerful tool from linear algebra. Finally, we explore ways to determine whether an image is a blur of another image using invariant functions.

2. Matrix Methods In Image Processing

A digital image is a two-dimensional discrete function, \(f(x,y)\) with \((x,y) \in \mathbb{Z}^2\), where each pair of coordinates is called a pixel. The word pixel derived from English “picture element” determines the smallest logical unit of visual information used to construct an image. Without loss of generality one image can be represented by a matrix where each element \(ij\) corresponds to the value of the pixel image position \((i,j)\). In Figure 1 a digital image is represented of size \(5 \times 12\)

![Fig. 1 Digital image with 5 × 12 pixels](image-url)
In the binary image \( f(x, y) \) each pixel is 0 or 1, i.e., the matrix that defines the image is a matrix with entries 0 or 1. In this case the image is set to white (1) and black (0). Figure 2(a) represents a binary image. A grayscale image also called monochrome only measures the light intensity (brightness). In this case, the value of \( f(x, y) \) for each pixel is an integer that measures the brightness of that pixel. The grayscale images vary in a range between \([0, NC - 1]\), where \( NC \) is the number of gray levels or intensity levels. The most common grayscale image using 8 or 16-bit per pixel, resulting in \( NC = 2^8 = 256 \) or \( NC = 2^{16} = 65536 \) distinct gray levels. Figure 2(b) represents a grayscale image with 8 bits. In a colour image the value of \( f(x, y) \) for each pixel measures the intensity of the colour at each pixel and it is represented by a vector with colour components. The most common colour spaces are RGB (Red, Green and Blue), HSV Hue, Saturation and Value) and CMYK (Cyan, Magenta, Yellow and black). In this paper the colour images are defined in RGB space. The three parameters of the RGB, which represent the intensities of the three primary colours of the model, define a three-dimensional space with three orthogonal directions (R, G, and B). Traditionally, the implementations of the model in the RGB graphics systems use integer values between 0 and 255 (8-bit). Figure 2(c) represents a color image.

Fig. 2 Representation of a binary, grayscale and colour image.

2.1 Matrix Operations

One of the first concepts that students get exposed to in a Linear Algebra course is matrix operations.

2.1.1 Addition

Let \( A \) and \( B \) be two matrices (images) with the same dimension, i.e. with the same number of rows and columns, say matrices. The matrix obtained by adding the previous matrices \( A \) and \( B \), called is shown in equation 2.1.

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix} + \begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1n} \\
b_{21} & b_{22} & \cdots & b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
b_{m1} & b_{m2} & \cdots & b_{mn}
\end{bmatrix}
\]
\[
\begin{bmatrix}
a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\
a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn}
\end{bmatrix}
\] (2.1)

Figure 3(c) depicts the addition of two colour images (a) and (b). In image of Figure 3(c) it can be observed the overlap of the two images.

2.1.2 Subtraction

The subtraction of two matrices A and B with the same size \((m \times n)\) can be defined by the equation 2.2

\[
A - B = A + (-1 \times B)
= \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}
- \begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1n} \\
b_{21} & b_{22} & \cdots & b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
b_{m1} & b_{m2} & \cdots & b_{mn}
\end{bmatrix}
= \begin{bmatrix}
a_{11} - b_{11} & a_{12} - b_{12} & \cdots & a_{1n} - b_{1n} \\
a_{21} - b_{21} & a_{22} - b_{22} & \cdots & a_{2n} - b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn}
\end{bmatrix}
\] (2.2)

The subtraction of one image with another image can be used to eliminate the background. For example, the subtraction of the image (b) to the image (a) in Figure 4 correspond to image (c) without background.
Fig 4. Subtraction the background to the image.

In Fig 5, the subtraction of grayscale image (a) to the 255 × I (255 × identity image) produces a negative image represented in (b)

2.1.3 Multiplication

Let A and B be matrices such that the number of columns of A is equal to the number of rows of B, say A is an m × p matrix and B is a p × n matrix. Then the product of the matrix A with the matrix B, called A × B is a m × n matrix whose ij element is obtained by multiplying ith row of A by jth column (Bij) of B [2, 10]. The product of two matrices is represented in equations 2.3.

\[
A \times B = \begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1p} \\
    a_{21} & a_{22} & \cdots & a_{2p} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{m1} & a_{m2} & \cdots & a_{mp} \\
\end{bmatrix}
\times
\begin{bmatrix}
    b_{11} & b_{12} & \cdots & b_{1n} \\
    b_{21} & b_{22} & \cdots & b_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    b_{p1} & b_{p2} & \cdots & b_{pn} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    c_{11} & c_{12} & \cdots & c_{1n} \\
    c_{21} & c_{22} & \cdots & c_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    c_{m1} & c_{m2} & \cdots & c_{mn} \\
\end{bmatrix}
\]

Where \(c_{ij}, i = 1, 2, \ldots, m \) & \(j = 1, 2, \ldots, n\) is defined by

\[
c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj}
\]

\[
= \sum_{k=1}^{p} a_{ik}b_{kj}
\]
With matrix multiplication it was possible to invert an image. In Figure 6 the original image \((a)\) was multiplied by \(I\) represented in equation 2.3

\[
I' = \begin{bmatrix}
0 & \ldots & 0 & 1 \\
\vdots & \ddots & \vdots & \vdots \\
\vdots & \ddots & \vdots & \vdots \\
1 & 0 & \ldots & 0
\end{bmatrix}
\]

In \((b)\) the matrix \(I'\) was multiplied at right of \(A\), and in \((c)\) at left.

\[
\begin{array}{c}
(a) \\
(b) \\
(c)
\end{array}
\]

**Fig 6 Multiply two images**

### 2.1.4 Element by element Multiplication

Sometimes it is useful applying multiplication element by element, i.e., multiply each element of \(A\) with the corresponding element of the matrix \(B\). Thus, if \(A\) and \(B\) are two matrices of dimension \(m \times n\) then the element by element multiplication \(A \times B\) is defined by equation 2.4.

\[
A \times B = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{bmatrix} \times \begin{bmatrix}
b_{11} & b_{12} & \ldots & b_{1n} \\
b_{21} & b_{22} & \ldots & b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
b_{m1} & b_{m2} & \ldots & b_{mn}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
a_{11} \times b_{11} & a_{12} \times b_{12} & \ldots & a_{1n} \times b_{1n} \\
a_{21} \times b_{21} & a_{22} \times b_{22} & \ldots & a_{2n} \times b_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} \times b_{m1} & a_{m2} \times b_{m2} & \ldots & a_{mn} \times b_{mn}
\end{bmatrix}
\]

\[(2.4)\]

A mask is a black image with the same size of the initial image where only the region of interest is set to white. This mask is applied to the initial image as a filter to only operate in the region of interest image, defined by the mask. A mask function is applied to the image using a multiplication element by element. The application of the mask \((b)\) of Figure 7 to the original image \((a)\) produces the image \((c)\).
The students explore the geometric applications with different values and they can analyze the use of geometric transformation using matrix notation.

### 2.1.5 Rotation

The rotation of an object from a positive theta angle (counter clockwise) around the origin is a geometric transformation that does not deform the object. To rotate an object it was necessary to turn all the pixels. The matrix homogeneous coordinates notation for the pixel rotation \( p(x; y) \) with theta angle is defined in polar coordinates by equation 2.5.

\[
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]  

(2.5)

In Figure 8, the rotations were made to correct the images. The image (a) was rotated \( \pi - \frac{\pi}{16} \) radians to produce the image (b) and image (c) was rotated \( -\frac{\pi}{16} \) radians to produce the image (d).
2.2 Geometric transformation to an arbitrary point

Transformation matrices can be applied consecutively to make compound movements and rotations to one image. To scale an object to an arbitrary point \( p = (x_1, y_1) \) it was necessary to perform the following elementary transformations:

1. Move the pixel \( p = (x_1, y_1) \) to the origin of the coordinate system, applying a translation \( T = (-x_1, -y_1) \) to all the pixels of the object.
2. Change the dimensions of the object by applying a scaling \( S = (s_x, s_y) \)
3. Move the pixel \( P = (x_1, y_1) \) to its original position, giving thus a back translation \( T = (x_1, y_1) \) to all pixels of the object

The matrix of the composite transformation \( T_c \) can be calculated by the equation 2.7

\[
T_c = T(x_1, y_1) \times S(s_x, s_y) \times T(-x_1, -y_1) \\
= \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix} \\
= \begin{bmatrix} s_x & 0 & x_1(1-s_x) \\ 0 & s_y & y_1(1-s_y) \\ 0 & 0 & 1 \end{bmatrix}
\] (2.7)

In Figure 9, the original image of the opera house in Sydney (a) was scaled by \( S(0.5,1) \) to the point \((36,30)\) to produce the image (b)

![Fig 9 Scaling to an arbitrary point](image)

2.3 Edge Detection

Students can study and explore the application of the gradient vector to the images. Edge detection in digital images is an important task used in many applications of image processing. An edge in the image represents a region where there is a sharp contrast, i.e. a rapid change of intensity. The main idea of the techniques for edge detection is the computation of the gradient of the image. The gradient of a function \( f(x, y) \) is defined by equation 4.10 whose amplitude is given by equation 2.9.

\[
\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}
\] (2.9)
\[ |\nabla f| = \sqrt{G_x^2 + G_y^2} \]

Often the magnitude of the gradient vector is approximated by the absolute value, i.e. the equation 2.10.

\[ |\nabla f| \approx |G_x| + |G_y| \quad (2.10) \]

A rapid change in intensity corresponds to a large value of the magnitude of the gradient, so the gradient of an image will have a high value on the edges of the image. A change of the intensity values is corresponding to the amplitude of the low gradient. Thus, the gradient of an image contains information on the edges in the image, and therefore it can be used to detect edges in an image.

A 3 × 3 region of the image represents the neighbourhood, where the centre point \( z_5 \) is designated by \( f(x, y) \) and the point \( z_1 \) by \( f(x - 1, y - 1) \) (see Fig 11).

| \( f(x - 1, y - 1) \) | \( f(x, y - 1) \) | \( f(x + 1, y - 1) \) |
| \( f(x - 1, y) \) | \( f(x, y) \) | \( f(x + 1, y) \) |
| \( f(x - 1, y + 1) \) | \( f(x, y + 1) \) | \( f(x + 1, y + 1) \) |

Fig 11. 3 × 3 region of an image centre in \( (x, y) \)

An approach to the first derivative of the function \( f(x, y) \) is defined by Prewitt operator with equations 2.12, where \( G_x \) represent the vertical operator and \( G_y \) the horizontal one.

\[
G_x = (z_7 + z_9 + z_8) - (z_1 + z_2 + z_3)
= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}
\]

\[
G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)
= \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}
\quad (2.12)
\]

A variation of these two equations using a weight of 2 for the central coefficients, giving some smoothness by the greater importance attached to the central point is called the Sobel operator, and it is represented in the equations 2.13.

\[
G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)
= \begin{bmatrix} -2 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
\]

\[
G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)
= \begin{bmatrix} -2 & 0 & 2 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}
\quad (2.13)
\]

There are also two Prewitt operators and two Sobel operators for detecting diagonal edges, which is to detect discontinuities in the diagonal directions. For diagonal edges using the Prewit operator, the equations are represented in 2.14.

\[
G_x = (z_2 + z_3 + z_6) - (z_4 + z_7 + z_9)
= \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}
\]

\[
G_y = (z_6 + z_8 + z_9) - (z_1 + z_2 + z_4)
\]

\[
G_x = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}
\]

\[
G_y = (z_6 + z_8 + z_9) - (z_1 + z_2 + z_4)
\]
and using the Sobel operator in equations 2.15

\[
G_x = (z_2 + 2z_3 + z_6) - (z_4 + 2z_7 + z_8)
\]

\[
= \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]

\[
G_y = (z_6 + z_8 + 2z_9) - (2z_4 + z_5 + z_4)
\]

\[
= \begin{bmatrix}
-2 & -1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]  

(2.15)

The application of these operators to the pixel \( P = (x, y) \) was defined by the equation 2.16,

\[
R = z_1f(x-1, y-1) + z_2f(x, y-1) + z_3f(x+1, y-1) + z_4f(x-1, y) + z_5f(x+1, y) + z_6f(x, y) + z_7f(x-1, y+1) + z_8f(x, y+1) + z_9f(x+1, y+1)
\]

(2.16)

The Figure 12 represents the edge detection of the original image (a). The image (b) is the horizontal edge detection, (c) the vertical edge detection and (d) the both detection using the Prewitt operator.

![Figure 12 the edge detection using Prewitt operator](image-url)

3. Image Processing Techniques

A gray scale digital image of size \( m \times n \), \( m, n \in N \) can be mathematically defined as a matrix with entries \( f(x, y), x = 0, 1, ..., m - 1, y = 0, 1, ..., n - 1 \), where the value \( f(x, y) \) represents the intensity or gray level of the image at the pixel \( (x, y) \). The intensity values of a gray-scale image range in a discrete interval between two numbers \( a \) and \( b \) (\( a < b \)) where \( a \) represents the lowest intensity (i.e black), \( b \) represents the highest intensity (i.e white) and all the values in between represent different levels of gray from black to white. The numbers \( a \) and \( b \) depend on the class of the image, a very commonly used range is the set of intensities \{0,1,2, ...,255\} where 0 represents black and 255 represents white.

Let \( f(x, y), x = 0, 1, ..., m - 1, y = 0, 1, ..., n - 1 \), represent an \( m \times n \) gray scale digital image. For the purposes of this discussion we assume that the intensity values range between \( a = 0 \) and \( b = 1 \)

3.1 Spatial Domain Techniques

Manipulations in the spatial domain are manipulations applied directly to the intensity values of the image (i.e \( f(x, y) \)). Examples include intensity transformations and spatial filtering, among others.
3.1.1 Intensity Transformations

Let \( T : [0,1] \rightarrow [0,1] \), consider the image represented by the \( m \times n \) matrix \( g(x,y) = T(f(x,y)), x = 0,1,...,m-1, y = 0,1,...,n-1 \). The function \( T \) is called intensity transformation and depending on its choice, the image of my result in an enhanced version of \( f \) for better human interpretation. Some examples of basic intensity transformations are:

3.1.1.1 Negative of an image

Inverts the intensity values of the pixels (ie sends black to white, white to black and inverts the gray scale values). \( T \) can be defined as:

\[
T(u) = 1 - u
\]

3.1.1.2 Gamma Transformation

Either maps a narrow range of dark input values into a wider range of output values, or maps a wide range of input values into a narrower range of output values. \( T \) can be defined as:

\[
T(u) = c(u + \alpha)^\gamma
\]

Where \( c, \alpha, \gamma \) are appropriate parameters.

3.1.1.3 Intensity Level Slicing

Highlight a specific range of intensities in an image. It can be done highlighting the desired range and leaving the rest of the image the same, or highlighting the desired range and changing everything else to a specific intensity. \( T \) can be defined by:

\[
T(u) = \begin{cases} 
I_1, & p \leq u \leq q \\
I_2, & \text{otherwise} 
\end{cases}
\]

Where \([p, q]\) is the desired range to highlight and \( I_1 \) and \( I_2 \) are appropriately chosen intensity values.

3.1.1.4 Spatial Filtering

Spatial filtering can be applied either to sharpen an image and increase detail or to smooth an image and reduce noise. There are two main types of spatial filtering: linear and nonlinear. Linear spatial filtering is carried out by applying a \( p \times q \) mask \( W \) to a neighborhood of each pixel in the image. These masks are often called filters and are applied through correlation or convolution.

An example of a linear filter is the averaging filter which changes the intensity values \( f(x, y) \) for the average intensity values of pixels in a neighbourhood of \((x, y)\). More precisely, the new intensity value at \((x, y)\) is given by:
\[
\frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x + s, y + t)
\]

Where \(a\) and \(b\) are fixed non-negative integers. This filter corresponds to a mask \(W\) of size \((2a + 1) \times (2b + 1)\) and with all entries equal to \(\frac{1}{(2a+1)(2b+1)}\). This process smooths the image and reduces noise.

An example of a non-linear filter is the first derivative of the image, used to sharpen the image and increase detail. The two first partial derivatives of an image can be defined as:

\[
\frac{\partial f}{\partial x}(x, y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \otimes f(x, y)
\]

\[
\frac{\partial f}{\partial y}(x, y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \otimes f(x, y)
\]

Where \(\otimes\) is the correlation defined by

\[
(W \otimes f)(x, y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} W(i, j)f(x + i, y + j)
\]

For \(p \times q\) matrix \(W\) with \(p = 2a + 1\) and \(q = 2b + 1\)

Then, the first derivative of \(f(x, y)\) is given by the image

\[
f'(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}(x, y)\right)^2 + \left(\frac{\partial f}{\partial y}(x, y)\right)^2}
\]

For \(x = 0, 1, ..., m - 1, \ y = 0, 1, ..., n - 1\)

4. Conclusion

All standard ways to manipulate images (matrix operations, geometric transformations, edge detection, etc.) may be performed by applying mathematical operations to the matrix associated with each image. A natural link between Linear Algebra and Digital Image Processing, supported by contemporary technologies and computational tools can be explored in elementary Linear Algebra courses. The concepts to be learned by the student must have meaning for the students, so that they can assimilate it. This means that teachers must discern the meaning of the concepts to the student. In this sense, the use of MATLAB software and the Image Processing can enrich teaching practice, improving student learning by exploring its resources in every activity proposed by the teacher. The students overcome the difficulties presented in the
study of Linear Algebra. Teachers are able to transmit the contents of Linear Algebra to students and teaching/learning process of Linear Algebra is more stimulating and motivating.

REFERENCES