

# The analysis and manipulation of a digitized image processing

V.Nethaji\* and A.P.Shanmugasundaram

Karpagam University, Coimbatore, Tamilnadu, India

Received: 22-January-2020; Revised: 27-February-2020; Accepted: 28-February-2020

©2020 V.Nethaji and A.P.Shanmugasundaram. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Abstract

*The digital image processing (DIP) is the process of digital images using various computer algorithms. This digital image processing has been employed in number of areas such as pattern recognition, remote sensing, image-sharpening, colour and video processing and medical. Since images are defined over two dimensions digital image processing may be modelled in the form of multi-dimensional systems. The generation and development of digital image processing are mainly affected by three factors: first one, development of computers; second, development of mathematics and third, demand for a wide range of applications including agriculture, military, industry and medical science has increased. When images are digitally stored in RAW format, they are saved in a matrix with each element representing the pixel's intensity value. These matrices are indexed in the form  $A(H, V)$ , where  $H$  is the pixel's horizontal position and  $V$  its vertical position.*

## Keywords

*Digital image processing (DIP), Compression, Cartesian, Gaussian, Matrix.*

## 1.Introduction

Image processing has become so deep-seated in everyday life that we often forget how quickly it has advanced over the last few periods. Today, we are able to snap photos and then rotate, crop, and apply various filters to the images directly on the device without any knowledge of photo editing techniques or digital image processing. Even when it comes to professional-position image processing, such as with aerial mapping, intelligent security, or medical imaging, both licensed and open source software packages exist to allow users without in-depth knowledge of the processing itself to simply feed in image data and generate ortho rectified maps, stitched composite views of a scene, or enhancing x-ray images [1]. This document goes behind the user interface and explores the digital representation of an image and how it is modified to create the end results many have become accustomed to receiving from these software packages.

The main goal of this work is to provide an intuitive understanding of the major techniques that have made a significant contribution to the image segmentation domain [2].

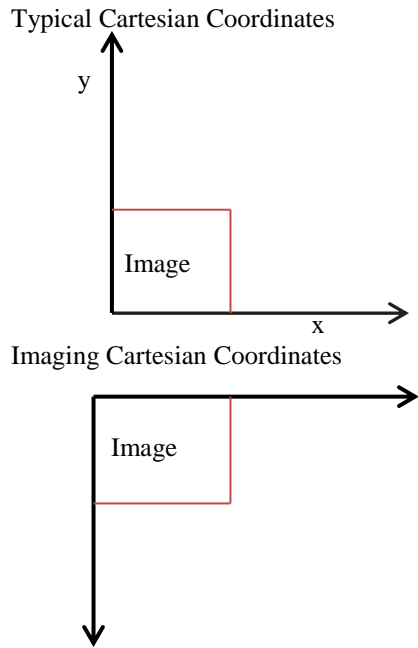
## 2.Image coordinate system

When referring to a specific pixel in an image, using the Cartesian coordinate system is an obvious choice since pixels are aligned in a grid. Each pixel is given an  $(x, y)$  coordinate to illustrate its position. However, in most cases, the origin  $(0,0)$  is not at the bottom left of the image. The general convention in image processing is to place the origin at the upper left corner of the image with positive values in  $x$  increasing from left to right and positive values in  $y$  increasing from top to bottom.

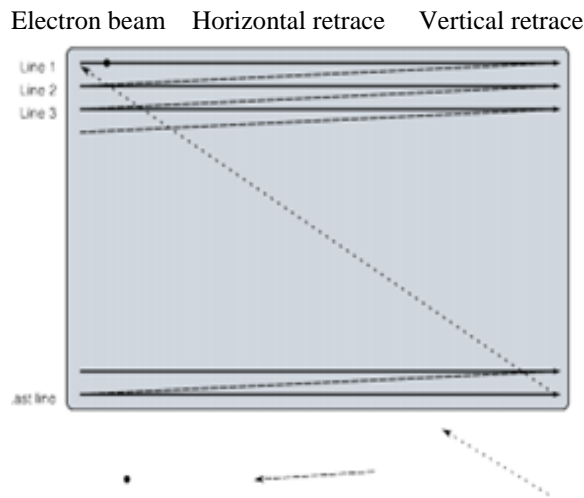
This convention dates back to CRT monitors, which swept a beam of electrons across the screen from left to right and top to bottom, mirroring how western languages are written and read. Furthermore, since images are often stored as matrices for image processing, this coordinate system is a better match to the indexing convention for this data structure [3].

Figure 1 shows the typical vs. imaging coordinate system. Figure 2 shows the scanning pattern.

\* Author for correspondence



**Figure 1** Typical vs. imaging coordinate system



**Figure 2** Scanning pattern

### 3. Matrices and images

In its rawest form, an image is saved to a matrix of dimensions equal to that of the sensor's resolution. For example, a 12-megapixel image would be saved to a matrix of 3000 elements tall by 4000 elements wide. When specifying the dimensions of a matrix, the height (y parameter) is given first, followed by the width (x parameter). Each element reports the pixel's value with a certain granularity depending on the bit depth. For illustrative purposes, the  $3 \times 4$  matrix below (named "A") will be used to represent the 12-megapixel image with a bit depth of 8-bits.

18

$$A = \begin{bmatrix} 64 & 183 & 59 & 161 \\ 129 & 251 & 42 & 207 \\ 5 & 198 & 12 & 32 \end{bmatrix}$$

Represented as an image, the matrix A would look like: Using the Cartesian coordinate system explained above, pixel (0,0) has the value of 65 and pixel (3,2) has the value of 33 [4]. Using matrix indices, the pixel with the value of 65 would be in element A (1,1), the pixel with value 33 would be in element A (3,4), and the pixel with value 207 would be in element A (2,4).

$$A = \begin{bmatrix} 64 & 183 & 59 & 161 \\ 129 & 251 & 42 & 207 \\ 5 & 198 & 12 & 32 \end{bmatrix}$$

Therefore, to transfer from Cartesian coordinates to matrix indices, the Cartesian coordinates need to each be increased by 1 and flipped:  $(x, y) = A(y+1, x+1)$ .

An algorithm based on amalgamation morphological watershed results with enhanced edge detection result. As a post handling step, to each of the segmented regions achieved, the color histogram algorithm was applied, improved the overall presentation of the watershed algorithm. The planned method enhanced the result of marker-controlled watershed for degraded images.

### 4. Image transformations matrix manipulations

Image transformations move images around without significant changes to the data contained in the image. The manipulations are performed on the pixel coordinates as opposed to the pixels themselves. A dot product between a transformation matrix and each  $(x, y)$  coordinate of the image produces a new set of coordinates to which existing pixels are mapped. In certain transformations, non-integer coordinates are generated, requiring some form of average pixel value from the original image to be mapped to the destination coordinate. Below are a few examples of simple image transformation matrices.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$2 \times 2$  scaling transformation matrix:

Scales the image size by factors.

A similar notation for this is to use a  $3 \times 3$  matrix as below. The addition of a "1" to the image coordinates does not impact the image itself, but allows for the addition of a translation component to the transformation matrix. *Figure 3* shows the original image. *Figure 4* shows the image scaled to 50%. *Figure 5* shows the image rotated clockwise 60 degrees



**Figure 3** Original image

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$3 \times 3$  scaling transformation matrix:  
Scales the image size by factors.



**Figure 4** Image scaled to 50%

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$3 \times 3$  scaling transformation matrix:  
Rotates the image clockwise by 0 degrees.



**Figure 5** Image rotated clockwise 60 degrees

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$3 \times 3$  translation transformation matrix:  
Moves the image by  $t$  pixels.

*Figure 6* shows the shifted image. *Figure 7* shows the result of simultaneous operation transformation matrix.

It presented an active method of digital image segmentation with the reducing over segmentation problem. A Gaussian  $7 \times 7$  mask was used for the smoothing purpose. It has applied to produce the final segmented image. It can be experimental from the final segmented images that the technique of reducing the over segmentation has been accomplished.



**Figure 6** Image shifted +8 px in y and +7 in X (Translation Highlighted)

These individual matrices can be combined to form a single transformation matrix to apply all the image transformations simultaneously [5]. The matrices are simply multiplied by one another using the dot product. Below is an example of a rotation, followed by a scaling, and this format of the matrix may be a little daunting at first, but once computed numerically all the terms scale down to numbers. For example, if the image were rotated by 60 degrees, scaled to 50% of its original size, and shifted 8 pixels down and 7 pixels to the right, the transformation matrix would be computed as:

$$\begin{bmatrix} 0.5\cos60 & -0.5\sin60 & 8.0.5\cos60 & -7.0.5\sin60 \\ 0.5\sin60 & 0.5\cos60 & 8.0.5\sin60 & +7.0.5\cos60 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & -0.43 & -1.03 \\ 0.43 & 0.25 & 5.21 \\ 0 & 0 & 1 \end{bmatrix}$$



**Figure 7** Result of simultaneous operation transformation matrix

Transformation matrices are used quite often behind the scenes in image processing [6]. For common

applications like Windows Paint, Adobe Photoshop to specialized Photogrammetry software such as Open CV and Pix4D, transformation matrices are used for simple and complex transformations alike. Image registration is a prime example of a complex transformation matrix. Methods currently exist to take two images captured simultaneously from two different cameras to determine a transformation matrix to align the image data, such that pixels with identical coordinates in each image refer to the same point on the ground. Registration can be used to align images captured from a mapping system in the air or from an intelligent security system on the ground.

There are existent several image improvement techniques that can improve the worth of digital images without making any loss to it. Some of the image improvement techniques comprise of contrast development, type, attentiveness, and fullness transformations, edge enhancement, and grey level slicing [7, 8]. Certain examples of development functions include contrast that gives to increasing the distinction between varied features within one picture, and spatial filtering to heighten the spatial image pattern. The method banks on the variation in brightness related to the edges containing some linear features. The key aim of this method is to alter certain components of an image to turn it into a more understandable image to a specific viewer and a given task. There are dissimilar techniques within this category and the techniques include contrast enhancement, hue, intensity, and saturation, edge enhancement, and Grey level slicing.

## 5. Image filtering

Filtering image data is done through matrix convolution. A square matrix of odd dimensions ( $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ , etc.) called the kernel is run through the image data matrix. The process involves an element wise multiplication of the kernel matrix with data from the image matrix. The results from each multiplication are summed and then saved to the coordinates that the centre of the kernel matrix occupies over the image data [2]. It is essentially a weighted sum of the nearest neighbours of each pixel. A few examples of kernel matrices are listed below.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

Edge detection      Sharpen Matrix      Gaussian Matrix

Applying the edge detection matrix (highlighted in green) to element A (2,2) from the Coordinates vs Indices section would look like this:

$$\begin{bmatrix} -1.65 & -1.182 & -1.57 & 163 \\ -1.125 & 8.250 & -1.43 & 207 \\ -1.4 & -1.199 & -1.11 & 33 \end{bmatrix}$$

The result for A'(2,2) would be 1314. As this is an 8-bit image, the value would be capped at 255, or white, indicating the presence of an edge. Since this is random data, also known as noise, edge detection is not practical in this scenario.

It used with 12 new and proposed chance structuring elements and morphological smoothing process to reduce the over segmentation problem. The basic attitude of using the structuring element in mathematical morphological operation lies in the fact that it serves as a seed or needle to collect the image information (2). The statistical analysis was shown and as per visual perception analysis the segmented images with proposed approach yields better accuracy in detection of edges and the over segmentation.

Below is an example of three different kernel matrices used to enhance an x-ray image. The first kernel matrix (a) performs averaging (to help remove noise), the second (b) performs edge enhancement, and the third (c) performs edge detection. For best results, these filters would be used in succession (Figure 8).

$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 10 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
(a)	(b)	(c)

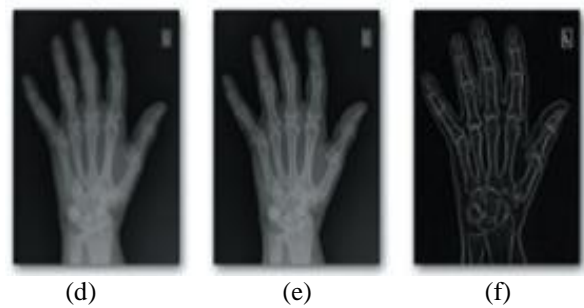


Figure 8 Matrices and x ray images

## 6. Image calculations

Performing calculations on image data can range from multiplying the image by a singular value to adding, subtracting, multiplying, or dividing images together. Here, the operation takes place on a pixel by pixel basis, so it is important that the information contained at the same coordinates in each image share a high correlation. For this reason, it is important to perform image registration before attempting to perform image calculations with images from two or more cameras. Multiplying two images together as matrices are known as the Hadamard product and is completed as follows:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \circ \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} a & 2b & 3c \\ 4d & 5e & 6f \\ 7g & 8h & 9i \end{bmatrix}$$

When performing these types of operations in certain image processing software, it is important to specify an image/matrix type that will allow for values higher than the specified bit depth as well as the possibility for negative numbers in the case of subtractions to ensure that no data is clipped [6]. In image compression using integer multi wavelets transform for the telemedicine applications. It suggested an effective compression and encoding presentation based on Integer multi wavelet transform of application. The proposed algorithm resulted in better quality images. The work focused on the implementation of lossless image data. They proposed multi wavelet-based compression for this problem, which had been shown to have much better coding efficiency and less computational complexity than existing approaches. This is how edge detection is performed for computer vision inspection applications. Finally, matrices of identical size can be added, subtracted, multiplied, and divided from each other in an element wise fashion to analyze the data based on certain mathematical models such as with precision agriculture.

## 7. Conclusion

Image recognition and its understanding are considered as an important subfield of artificial intelligence. Digital image processing deals with manipulation of digital images through a digital computer. When images are digitally stored in RAW format, they are saved in a matrix with each element representing the pixel intensity value.

These matrices are indexed in the form A (H, V), where H is the pixel's horizontal position and V its vertical position. Contrary to the typical origin location in Cartesian coordinates, the origin of an image file is the upper left-hand corner of the image with values increasing horizontally from left to right along the x axis and from top to bottom for the y axis. This helps to ensure that the origin of the image in Cartesian coordinates lines up with the indexing method used for matrices. The containerized design allows future updates to spiders to take place without concern regarding the underlying hardware.

## Acknowledgment

None.

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References

- [1] Almohri H, Gray JS, Alnajjar H. A real-time DSP-based optical character recognition system for isolated arabic characters using the TI TMS320C6416T (Doctoral dissertation, University of Hartford).
- [2] Erickson BJ, Korfiatis P, Akkus Z, Kline T, Philbrick K. Toolkits and libraries for deep learning. *Journal of Digital Imaging*. 2017; 30(4):400-5.
- [3] Anagnostopoulos CN, Anagnostopoulos IE, Loumos V, Kayafas E. A license plate-recognition algorithm for intelligent transportation system applications. *IEEE Transactions on Intelligent transportation systems*. 2006; 7(3):377-92.
- [4] Puri C, Singh S. Image segmentation and classification-a review. *International Journal of Innovative Research in Science, Engineering and Technology*. 2014.
- [5] Taujuddin M, Afifi NS, Lockman NA. Image compression using wavelet algorithm. *International Seminar on the Application of Science and Mathematics*. 2011.
- [6] Deodhare D, Suri NR, Amit R. Preprocessing and image enhancement algorithms for a form-based intelligent character recognition system. *International Journal of Computer Science and Applications*. 2005; 2(2):131-44.
- [7] Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. *International Journal of Computer Vision*. 2004; 59(2):167-81.
- [8] Pandit H, Shah DM. Application of digital image processing and analysis in healthcare based on medical palmistry. In *international conference on intelligent systems and data processing 2011* (pp. 56-9).



**Dr. V.Nethaji**, Tamilnadu, India, 01 November 1981. I received Bachelor of Science (B.Sc) in Computer Technology (Regular) from K..S.Rangasamy College of Technology, Tiruchengode, Tamilnadu, India in 2003. Master of Computer Applications (M.C.A) in Computer Application (Regular) from Periyar University, Salem, Tamilnadu, India in 2006. Master of Philosophy (M.Phil) in Computer Science (Part-Time) from Periyar University, Salem, Tamilnadu, India in 2008. Doctorate (PhD) from Karpagam University, Coimbatore, Tamilnadu, India in April 2016 and the title is "AN approach to detect software anomalies using pragmatic bayes and fault localization in cloud environment".  
Email: nethaj.babu@gmail.com



**Dr. A. P. Shanmugasundaram**, Tamilnadu, India, 30 October 1980. I received Bachelor of Science (B.Sc) in Computer Technology (Regular) and Master of Computer Application from K.S.Rangasamy College of Technology, Tiruchengode, Tamilnadu, India in 2003 & 2006. Master of Philosophy (M.Phil) in Computer Science (Part-Time) from Periyar University, Salem, Tamilnadu, India in 2008. Doctorate (PhD) from Karpagam University, Coimbatore, Tamilnadu, India in April 2016 and the title is "Reliable mulicast routing and multipath data delivery using spatial time scheduling technique in ad hoc network".