University of Diyala
Computer Science Department
Image Processing
3rd Class
Lecturer: Dr. Jumana Waleed Salih

- -

Image Processing

معالجة صور

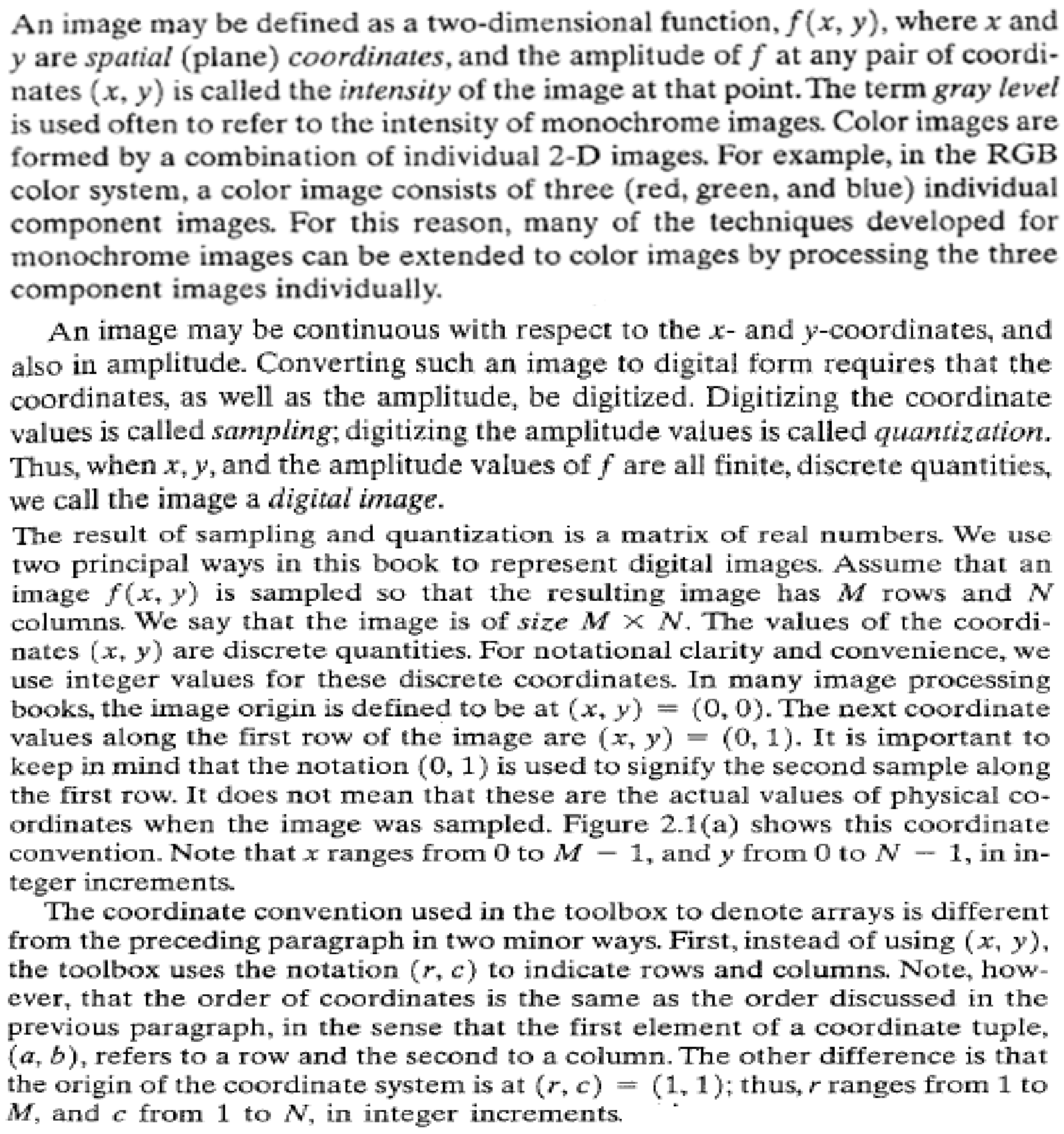
1st lecture

Introduction

Interest in digital image processing methods stems from two aspects:

1) Improve images for human interpretation;
2) Process images; include storage, transmission and representation.

What is a digital image?

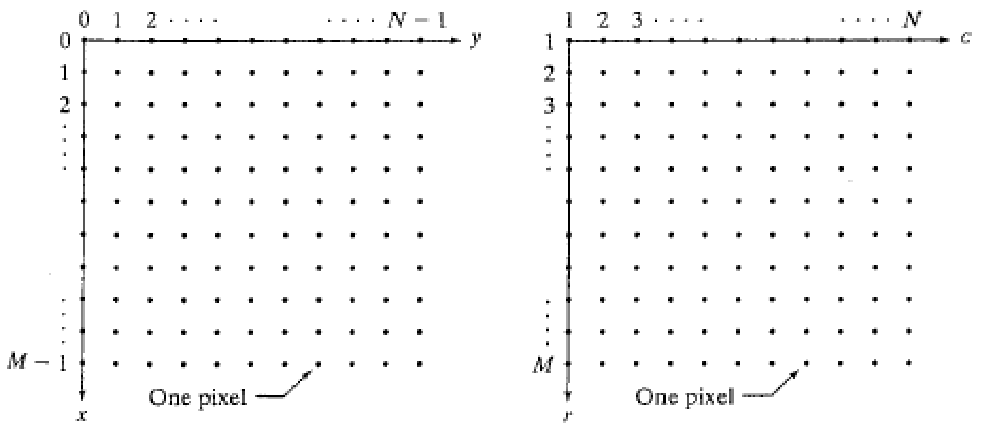


An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are *spatial* (plane) *coordinates*, and the amplitude of f at any pair of coordinates (x, y) is called the *intensity* of the image at that point. The term *gray level* is used often to refer to the intensity of monochrome images. Color images are formed by a combination of individual 2-D images. For example, in the RGB color system, a color image consists of three (red, green, and blue) individual component images. For this reason, many of the techniques developed for monochrome images can be extended to color images by processing the three component images individually.

An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. Converting such an image to digital form requires that the coordinates, as well as the amplitude, be digitized. Digitizing the coordinate values is called *sampling*; digitizing the amplitude values is called *quantization*. Thus, when x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a *digital image*.

The result of sampling and quantization is a matrix of real numbers. We use two principal ways in this book to represent digital images. Assume that an image $f(x, y)$ is sampled so that the resulting image has M rows and N columns. We say that the image is of *size* $M \times N$. The values of the coordinates (x, y) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at $(x, y) = (0, 0)$. The next coordinate values along the first row of the image are $(x, y) = (0, 1)$. It is important to keep in mind that the notation $(0, 1)$ is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure 2.1(a) shows this coordinate convention. Note that x ranges from 0 to $M - 1$, and y from 0 to $N - 1$, in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (x, y), the toolbox uses the notation (r, c) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate tuple, (a, b), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at $(r, c) = (1, 1)$; thus, r ranges from 1 to M, and c from 1 to N, in integer increments.



a b

Coordinate conventions used (a) in many image processing books, and (b) in the Image Processing Toolbox.

The coordinate system in this figure leads to the following representation for digitizing image function:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

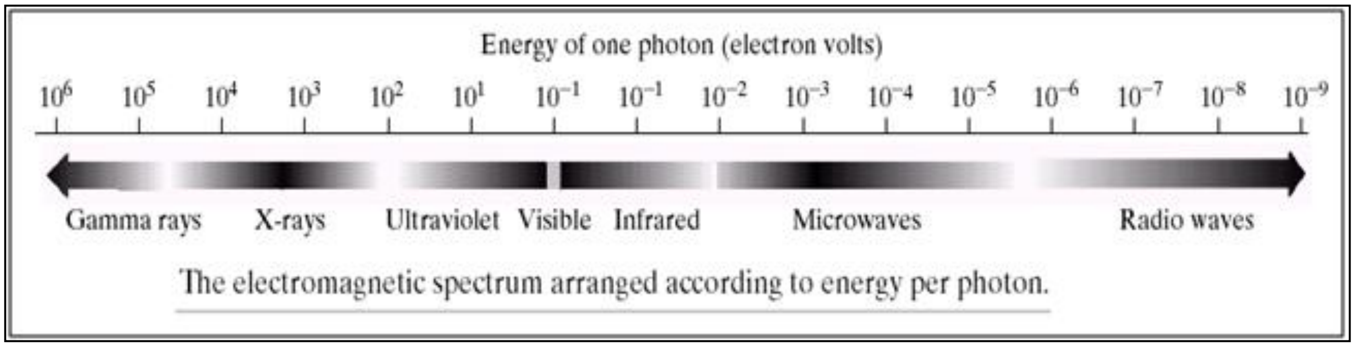
The right side of this equation is a digital image by definition. Each element of this array is called an *image element*, *picture element*, *pixel*, or *pel*. The terms *image* and *pixel* are used throughout the rest of our discussions to denote a digital image and its elements.

Levels of image processing

1. Low-level processes: involve primitive operations such as reduce noise, contrast enhancement, image sharpening and so on. Usually it is a procedure from images to images.
2. Mid-level processes: involve image segmentations (partition an image into regions and objects), image recognition (tell what objects are in an image) and so on. Usually its input is an image but outputs are attributes extracts from the image such as edges, contours, and the identity of individual objects).
3. Higher-level process: involve “making sense” of an ensemble of recognized objects.

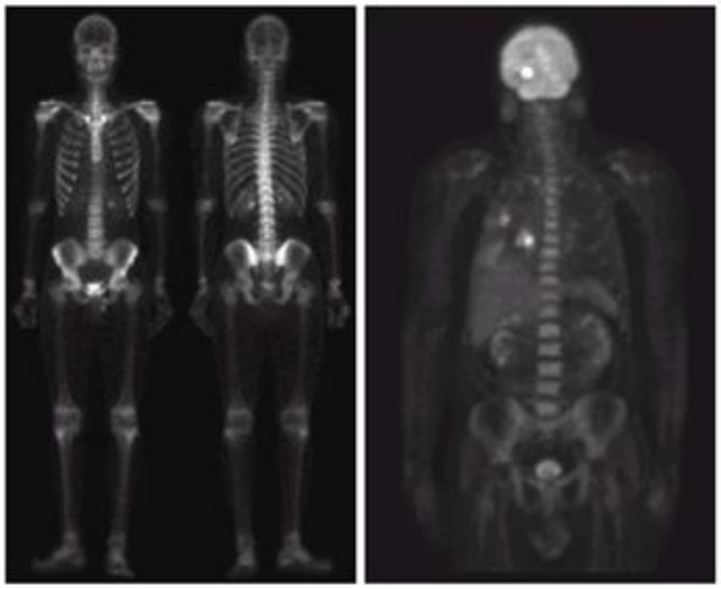
Examples of Fields that Use Digital Image Processing

It is one of the simplest ways to categorize images according to their resource (e.g., visual, X-ray, and so on). The spectrum below ranges from gamma rays (highest energy) at one end to radio waves (lowest energy) at the other.



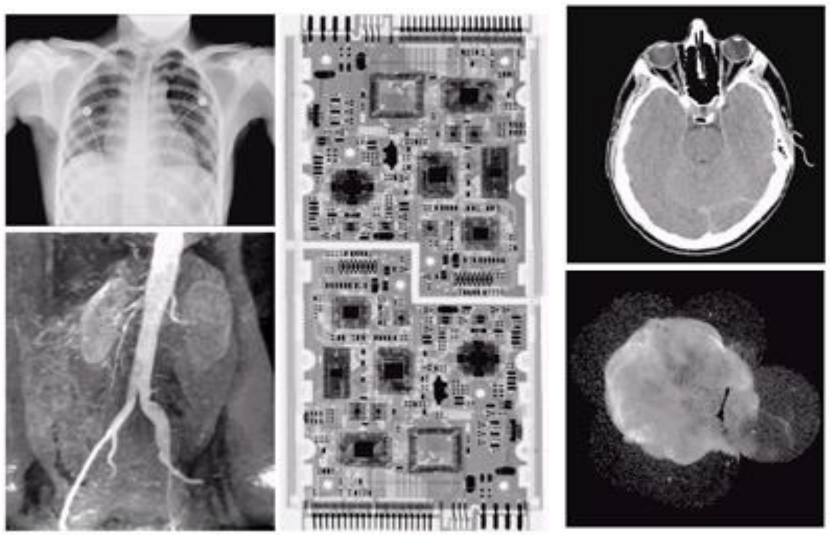
Gamma-Ray Imaging

Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations.



X-Ray Imaging

The best known use of X-rays is medical diagnostics, but they also are used in industry and other areas, like astronomy.



Imaging in the Ultraviolet Band

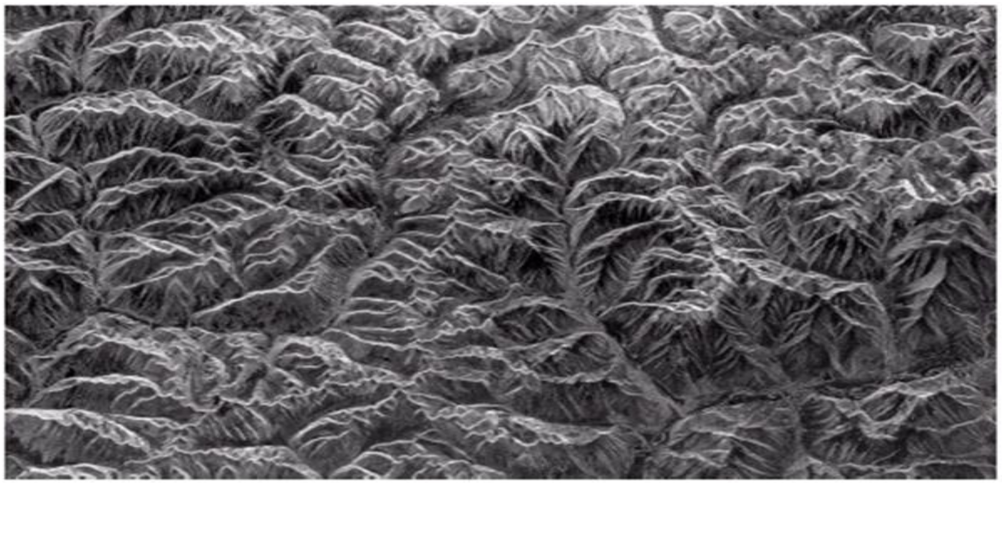
Applications of ultraviolet “light” include lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations.

Imaging in the Visible and Infrared Bands

The visual band of the electromagnetic spectrum is the most familiar in our activities. So their applications are also the widest compared with other bands. And infrared band is often viewed visual light. Some applications of the visual light are light microscopy, astronomy, remote sensing, industry, and law enforcement.

Imaging in the Microwave Band

A typical application in the microwave band is radar.



Spaceborne radar image of mountains in southeast Tibet. (Courtesy of NASA.)

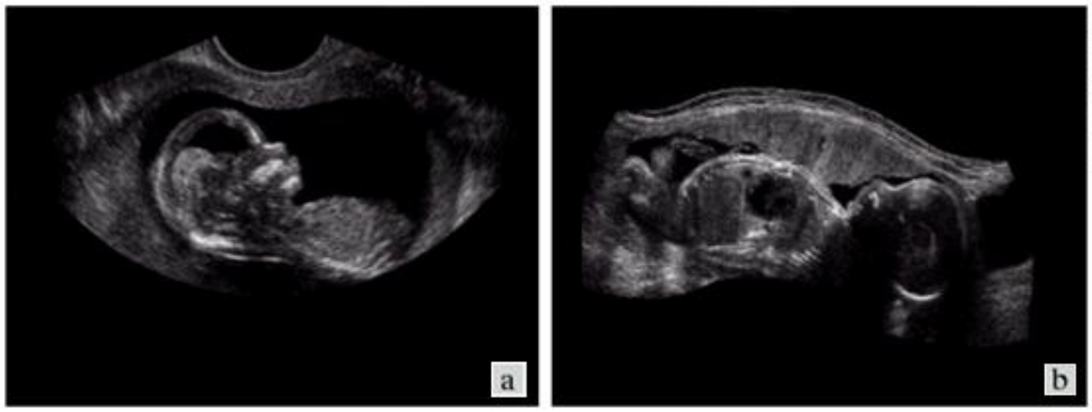
Imaging in the Radio Band

The major applications in the radio band are in medicine and astronomy.

Examples in which Other Imaging Modalities Are Used

Ultrasound imaging

The best known applications of the ultrasound imaging are in medicine, especially in obstetrics.



Examples of ultrasound imaging. (a) Baby. (2) Another view of baby.

The MATLAB Working Environment

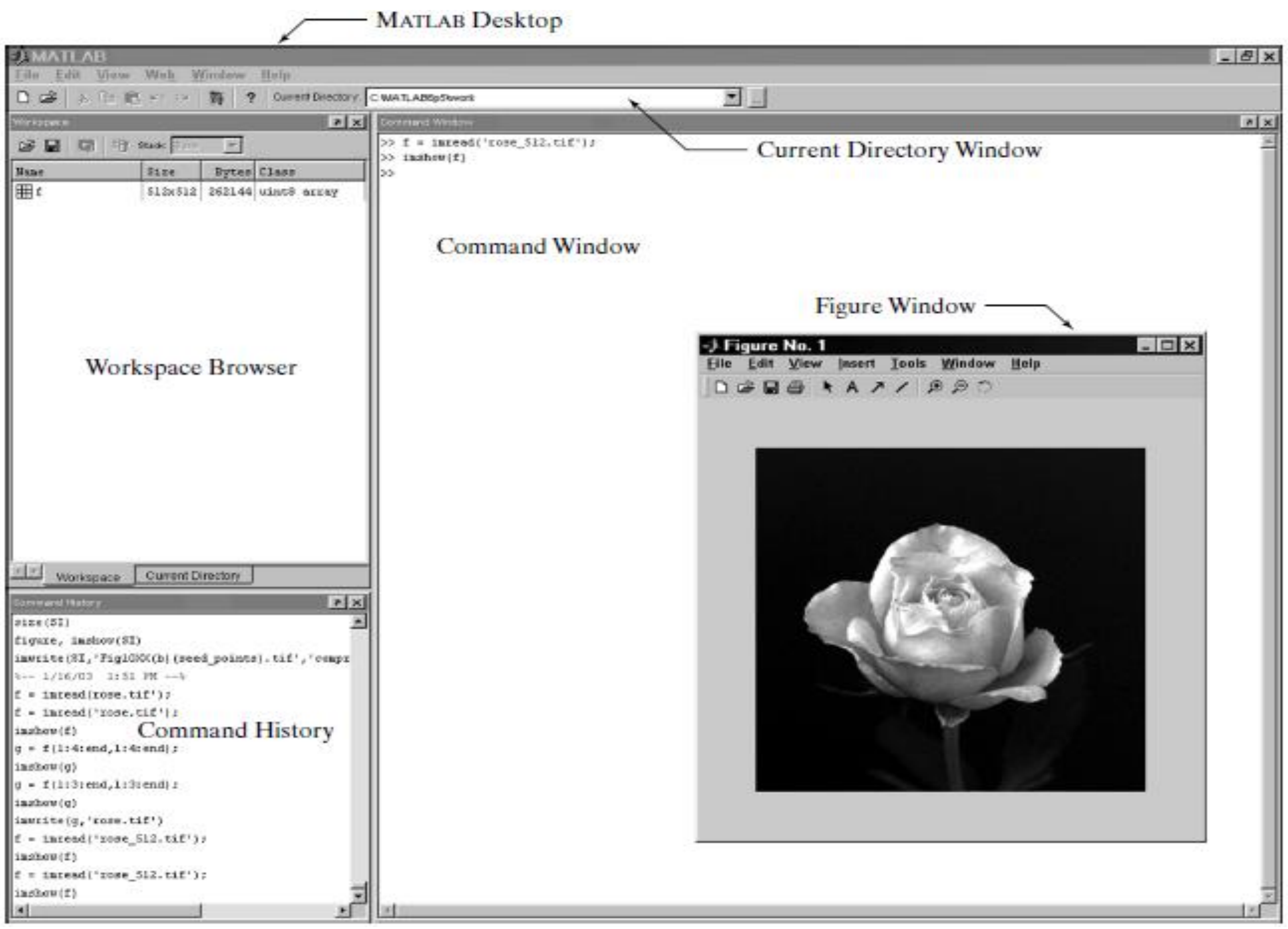
In this section we give a brief overview of some important operational aspects of using MATLAB.

The MATLAB Desktop

The MATLAB desktop is the main MATLAB application window. As figure below shows, the desktop contains five subwindows: the Command Window, the Workspace Browser, the Current Directory Window, the Command History Window, and one or more Figure Windows, which are shown only when the user displays a graphic.

The Command Window is where the user types MATLAB commands and expressions at the prompt (>>) and where the outputs of those commands are displayed.

MATLAB defines the workspace as the set of variables that the user creates in a work session. *The Workspace Browser* shows these variables and some information about them. Double-clicking on a variable in the Workspace Browser launches the Array Editor, which can be used to obtain information and in some instances edit certain properties of the variable.



```
>> f = imread('rose_512.tif');
>> imshow(f)
>>
```

Name	Size	Bytes	Class
f	512x512	262144	uint8 array

```
size(SI)
figure, imshow(SI)
imwrite(SI,'Fig10XX(b)(seed_points).tif','compr
%-- 1/16/03  1:51 PM --%
f = imread(rose.tif');
f = imread('rose.tif');
imshow(f)
g = f(1:4:end,1:4:end);
imshow(g)
g = f(1:3:end,1:3:end);
imshow(g)
imwrite(g,'rose.tif')
f = imread('rose_512.tif');
imshow(f)
f = imread('rose_512.tif');
imshow(f)
```

The MATLAB desktop and its principal components

The Current Directory tab above the Workspace tab shows the contents of the current directory, whose path is shown in the Current Directory Window. For example, in the Windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “Work” is a subdirectory of the main directory “MATLAB,” which is installed in drive C. Clicking on the arrow in the Current Directory Window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory. MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories in the computer file system.

The Command History Window contains a record of the commands a user has entered in the Command Window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the Command History Window by right-clicking on a command or sequence of commands.

Images as matrices

A digital image can be represented naturally as a MATLAB matrix:

$$\mathtt{f} = \begin{bmatrix} \mathtt{f(1,1)} & \mathtt{f(1,2)} & \cdots & \mathtt{f(1,N)} \\ \mathtt{f(2,1)} & \mathtt{f(2,2)} & \cdots & \mathtt{f(2,N)} \\ \vdots & \vdots & & \vdots \\ \mathtt{f(M,1)} & \mathtt{f(M,2)} & \cdots & \mathtt{f(M,N)} \end{bmatrix}$$

where `f(1, 1)` $= f(0, 0)$ (note the use of a monospace font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation `f(p, q)` denotes the element located in row `p` and column `q`. For example, `f(6, 2)` is the element in the sixth row and second column of the matrix `f`. Typically we use the letters `M` and `N`, respectively, to denote the number of rows and columns in a matrix. A $1 \times N$ matrix is called a *row vector*, whereas an $M \times 1$ matrix is called a *column vector*. A 1×1 matrix is a *scalar*.

Matrices in MATLAB are stored in variables with names such as `A`, `a`, `RGB`, `real_array`, and so on. Variables must begin with a letter and contain only letters, numerals, and underscores.

Reading Images

```
imread('filename')
```

For example:

```
>> f = imread('D:\myimages\chestxray.jpg');
```

Format Name	Description	Recognized Extensions
TIFF	Tagged Image File Format	`.tif`, `.tiff`
JPEG	Joint Photographic Experts Group	`.jpg`, `.jpeg`
GIF	Graphics Interchange Format[†]	`.gif`
BMP	Windows Bitmap	`.bmp`
PNG	Portable Network Graphics	`.png`
XWD	X Window Dump	`.xwd`

[†] GIF is supported by `imread`, but not by `imwrite`.

TABLE 1 Some of the image/graphics formats supported by `imread` and `imwrite`, starting with MATLAB 6.5. Earlier versions support a subset of these formats. See online help for a complete list of supported formats.

Function size gives the row and column dimensions of an image:

```
>> size(f)
ans =
    1024 1024
```

This function is particularly useful in programming when used in the following form to determine automatically the size of an image:

```
>> [M, N] = size(f);
```

This syntax returns the number of rows (M) and columns (N) in the image.

The whos function displays additional information about an array. For instance, the statement

```
>> whos f
```

gives

```
Name        Size             Bytes        Class
f           1024x1024        1048576      uint8 array
Grand total is 1048576 elements using 1048576 bytes
```

Displaying images

Images are displayed on the MATLAB desktop using function imshow, which has the following syntax:

imshow (*f*)

If another image, g, is displayed using imshow, MATLAB replaces the image in the screen with the new image. To keep the first image and output a second image, we use function figure as follows:

```
>> figure, imshow(g)
```

Using the statement

```
>> imshow(f), figure, imshow(g)
```