Lecture 1

Fourth stage

Medical Physical Department

Medical Imaging Processing

Introduction to Digital Image Processing

By

Asst. Prof. Dr. Mehdi Ebady Manaa

Introduction to Digital Images

Digital Image

The International Standards Organization (ISO) defines computer graphics as the sum total of "methods and techniques for converting data for a graphics device by computer." This definition would probably not help a reader totally unfamiliar with the field to understand what it's all about. In fact, the best way to understand a field is to grasp what its main problems are. From this point of view, the ISO definition can be said, with goodwill, to define the main problem of computer graphics: converting data into images. The process of converting data into images is known as visualization. It is schematically illustrated in Figure 1.1. In order to understand computer graphics, then, we must study the methods for creating and structuring data in the computer as well as methods for turning these data into images. These two steps correspond to the two main areas of research in computer graphics: modeling and visualization. In this book we will not study modeling or data visualization. Instead, we will focus on a more fundamental and very important problem: understanding the notion of an image and also the techniques of image manipulation by computer—in other words, image processing. At the same time, this is not a typical image processing book, because it covers primarily the aspects of image processing used most often in computer graphics

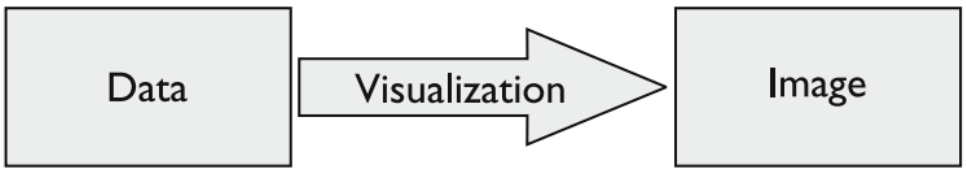


Fig. 1.1. Computer graphics: converting data into images.

Image processing used most often in computer graphics. Since its inception, computer graphics has sought methods to allow the visualization of information stored in computer memory. Since there are practically no limitations on the nature or origins of such data, researchers and professionals use computer graphics today in the most diverse fields

of human knowledge. Its use is important whenever it is necessary to have a visual representation of objects, actions, relations, or concepts. The importance of this visual representation is reflected in the huge number of computer graphics applications, ranging from scientific visualization to special effects in the entertainment industry. Partly because computer graphics has so many applications, there are no sharp boundaries between it and related fields. However, we can take as a working criterion in differentiating among these fields the nature of the input and output of the process in question, as shown in Figure 1.2. In data processing, the system takes in data and, after processing, returns data of more or less the same nature. For example, a bank account management system processes input transactions and yields output data such as a daily balance, interest earned, and so on. In computer graphics, the input data are (typically) nonvisual, and the output is an image that can be seen through some graphics output device. For instance, the account management system of the preceding paragraph might plot a graph of the daily balance over a period.

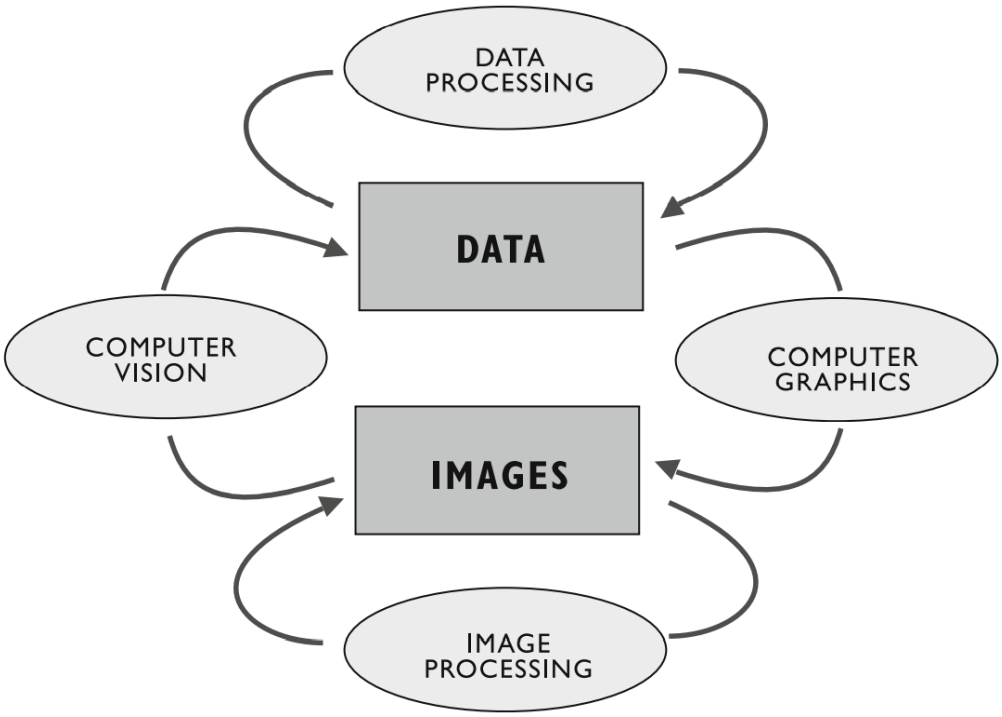
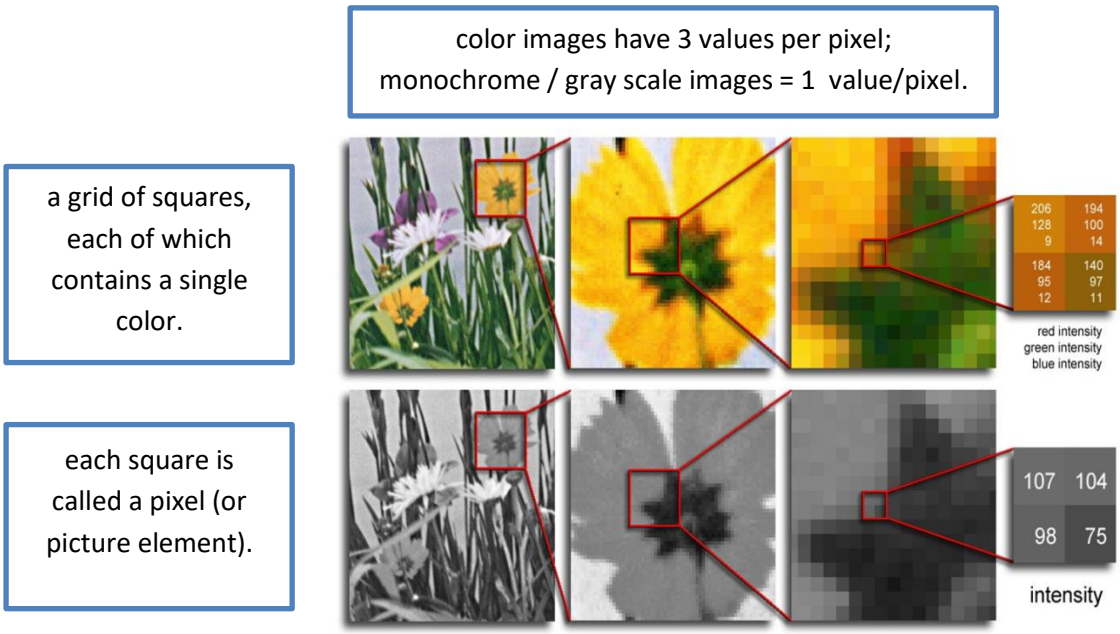


Fig. 1.2. Computer graphics and kindred disciplines.

Figure 1.3 shows the position of pixels in the images.



Pixels

- A digital image, I, is a mapping from a 2D grid of uniformly spaced discrete points {p = (r,c)}, into a set of positive integer values, {I(p)} or a set of vector values e.g., $\{[R\ G\ B]^T(p)\}$.
- Each column location of each row in I has a value.
- The pair (p, I(p)) is a "pixel" (for picture element).
- p = (r,c) $\longrightarrow$ pixel location indexed by row r & column c.
- I(p) = I(r,c) $\rightarrow$ Value of the pixel at location p.
- If I(p) is a single number $\rightarrow$ I is monochrome (B&W).
- If I(p) is a 3 element vector $\longrightarrow$ I is a colour (RGB) image.

Monochromatic Case:

- We call the values at each pixel intensities
- Smaller intensities denote a darker pixel
- Bigger intensities denote a lighter pixel

Colour Case:

- Think of a colour image as a 3D matrix ⟶ First layer is red, second layer is green, third layer is blue.
- Why RGB? All colours found in nature can naturally be decomposed into Red, Green and Blue ⟶ This is basically how CCD cameras work!.
- The three element vector tells you how much red, green and blue the pixel is compromised of (i.e. $[R\ G\ B]^T = [0\ \ 255\ 0]$ ⟶ No red, no blue, all green.

Questions

- How do digital cameras take images (very basic)?

  *Uses sampling and quantization

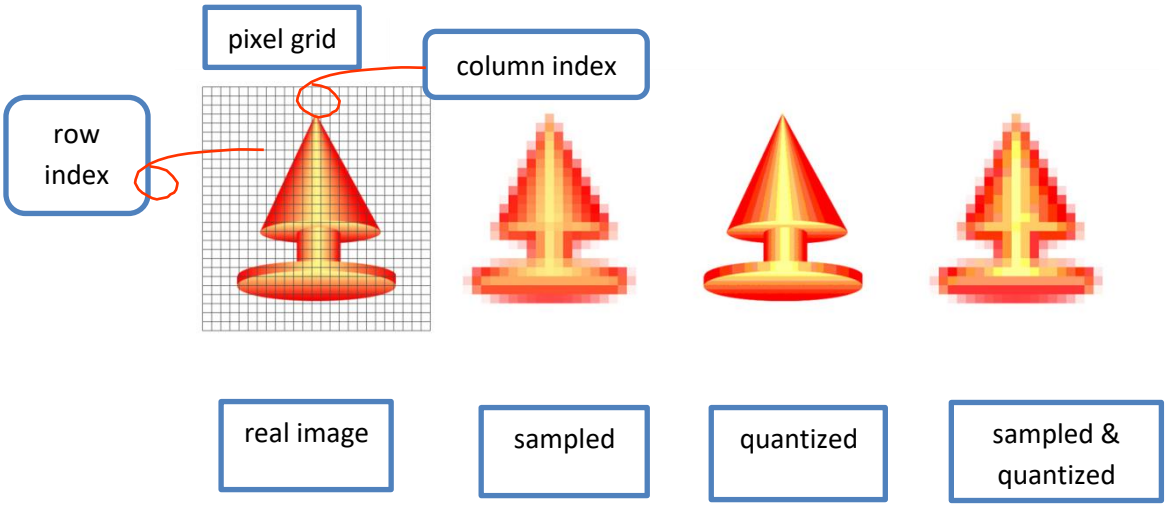
- What we see now through our eyes is continuous?

  * There is essentially an infinite amount of points that comprise our field of view (FoV)

  *Not good, because we want to store this information!

- We first need to sample the FoV ⟶ Transfer the FoV to a rectangular grid, and grab the colour in each location of the grid.

Sampling and Quantization



- We're not done yet! There are also an infinite number of possible colours.
- We will now need to *quantize* the colours.
- Quantizing will reduce the total number of colours to a smaller amount.
- Key $\longrightarrow$ Quantize accurately so that we can't tell much difference between the original image and the quantized one.
- A digital image is essentially taking our FoV and performing a sampling and quantization.
- Values are now discrete and positive.

Digital Images Characteristics

- Digital images store their intensities / colour values as

  discrete and positive values.

- Usually, digital images need 8 bits for B & W and 24- bits for colour (8 bits for each primary colour).
- B & W – 0 for Black and 255 for White $\longrightarrow$ All integers.
- Colour – 0 to 255 for Red, Green and Blue $\longrightarrow$ All integers.

Note: We can consider a colour image as three 2D images.

- Without compression, files would be very large!
- Compression algorithms (PNG, JPEG, etc.) eliminate extra information to reduce the size of the image.

R/W Images in MATLAB

- So we have an image file …how do I access the info?
- Open up MATLAB and change working directory to where image is stored.
- Use the imread ()function
- im = imread('name_of_image.ext')
- Use single quotes ,and type in the full name of the image with its extension(bmp, jpg, etc).

- im will contain a 2D matrix (rows x cols) of B&W values or a 3D matrix (rows x cols x3) of colour values.
- Matrix corresponds to each pixel in the digital image for B & W or a colour component of a pixel in colour.

— How do I access a pixel in MATLAB à B&W case?

- pix = im(row, col),
- row & col :Row & column of the pixel to access.
- pix contains the intensity value.
- Access elements in an array by round braces ,not square!.
- For you C buffs → Indexing starts at 1, not 0.

How do I access a pixel in MATLAB à Colour case?

- pix = im (row,col,1) → Red colour value
- pix = im (row,col,2) → Green colour value
- pix = im (row,col,3) → Blue colour value
- 3rd argument → 3rd dimension of matrix
- Only grabs one colour value at a time!

How can I get the RGB pixel entirely? Use the command

- pix = im (row, col, :);
- means to grab all values of one dimension.
- However, this will give you a 1 x 1 x 3 matrix… we just want an array! Call the squeeze() command.
- pix = squeeze (im (row ,col,:);
- Now a 3 x 1 vector. To access R, G and B values, do:
- red = pix(1) →Red, gr = pix(2) →Green,
- blue = pix(3) →Blue

—So I know how to get pixels ;how can I modify them in the image?

- Easy! Just go backwards
- For a B & W Image do:
  im(row, col)= pix;
- For a colour image ,do either:
  im(row,col,1) = red;
  im(row,col,2) = green;

```
im(row,col,3) = blue;  or
im(row, col,:) = [red; green; blue] or
im(row, col,:) = rgb; % rgb - 3 x 1 vector.
```