Chapter 1

Introduction

1.1 Notation

Images:

An $M \times N$ image in 256 grey scale (or grey levels) is a matrix

$$A = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0N} \\ a_{10} & a_{11} & & \\ \vdots & & & \ddots \\ a_{M0} & a_{M1} & \dots & a_{MN} \end{bmatrix}$$

also expressed as $\{a_{jk}\}, j = 0, 1, ...(M-1), \;\; k = 0, 1, ..., (N-1)$, and the elements are integers, $a_{jk} \in \{0, 1, 2, ..., 255\}$. Images are also viewed as functions of two variables, $f(x, y)$, where the coordinate pair (x, y) denotes some point on the 2D plane and $f(x, y)$ is the intensity of the image at that point.

Display:

Images are displayed as a rectangular array of grey squares. Each grey square is called a "pixel", and the greyness depends on the value of a_{jk} in the matrix. For $a_{jk} = 0$, the pixel is black, for $a_{jk} = 255$, the pixel is white and for values in between various shades of grey are used.

When viewed as a function of two variables, images may be represented as a surface. However this is a mathematical convenience usually used for image filters. It normally does not make sense to display an image of a scene as a surface.

Types of images:

The three types of images that will be used often in this presentation, are

— Binary images, where $a_{jk} \in \{0,1\}$.

— Grey scale images, where $a_{jk} \in \{0,1,2,...,255\}$.

— Normalized grey scale images, where $a_{jk} \in [0,1]$.

— Colour images, where each pixel is displayed in colour using three values (r_{jk}, g_{jk}, b_{jk}), the so-called RED, GREEN and BLUE colour channels. These triplets are usually packed into three different matrices,

$$\begin{aligned} R &= \{r_{jk}\}, j = 0,1,...,N, \quad k = 0,1,...,M, \\ G &= \{g_{jk}\}, j = 0,1,...,N, \quad k = 0,1,...,M, \\ B &= \{b_{jk}\}, j = 0,1,...,N, \quad k = 0,1,...,M, \end{aligned}$$

These matrices are referred to as the RED, GREEN and BLUE "colour planes" respectively.

The pixels in a binary image are often said to be "on" when the $a_{jk} = 1$ and "off" when $a_{jk} = 0$.

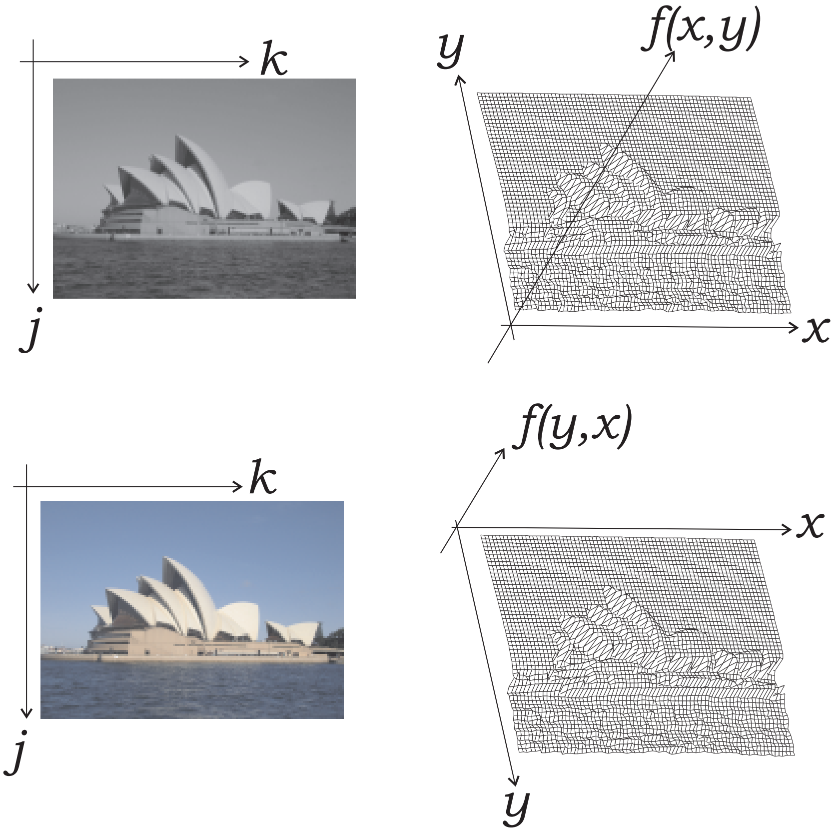

Figure 1. TOP LEFT: A grey scale image, BOTTOM LEFT: A colour image, TOP RIGHT: Viewed as a surface plot with conventional direction for axes. BOTTOM RIGHT: A surface plot but with axes directions more convenient for images.

<u>Pixels:</u>

Pixels are in effect positions on the plane. The convention to denote a single pixel position by a single symbol will often be employed, for example

$$\mathbf{p} = (j, k), \quad \mathbf{q} = (j + 1, k),$$

where $\mathbf{q}$ is the pixel "below" $\mathbf{p}$.

<u>Directions in the 2D-plane:</u>

Images are normally displayed the way matrices are written, first index j running vertically down, then the second index k running horizontally across.

When viewed as a function one normally lets x run horizontally to the right, and lets y run vertically up. This notion clashes with the ordering of the matrix indices. When working with images, it is perhaps more convenient to consider the function $f(y, x)$, where y is associated with j and x is associated with k, *and* in addition one inverts the y axis so that its positive part runs downwards (but we keep the positive part of the x-axis running to the right).

1.2 Pixel Neighbourhoods

Processing of images often consists of performing operations on a pixel $\mathbf{p} = (j, k)$, depending on the values of its neighbours. Some often used neighbourhoods are as follows:

<u>4-adjacency:</u>

The *4-neighbourhood* of pixel $\mathbf{p} = (j, k)$ is a set of pixels

$$N_4(\mathbf{p}) = \{(j + 1, k), (j - 1, k), (j, k + 1), (j, k - 1)\}.$$

If $\mathbf{q} \in N_4(\mathbf{p})$ then it is said that $\mathbf{q}$ is *4-adjacent to* $\mathbf{p}$. Note the commutativity of this relation: if $\mathbf{q} \in N_4(\mathbf{p})$ then $\mathbf{p} \in N_4(\mathbf{q})$ and the reverse is also true.

<u>D-diagonal-adjacency:</u>

The *diagonal-neighbourhood* of pixel $\mathbf{p} = (j, k)$ is a set of pixels

$$N_D(\mathbf{p}) = \{(j + 1, k + 1), (j - 1, k + 1), (j - 1, k + 1), (j - 1, k - 1)\}.$$

If $\mathbf{q} \in N_D(\mathbf{p})$ then it is said that $\mathbf{q}$ is *D-adjacent to* $\mathbf{p}$. D-adjacency is also commutative.

<u>8-adjacency:</u>

The *8-neighbourhood* of pixel $\mathbf{p} = (j, k)$ is a set of pixels

$$N_8(\mathbf{p}) = N_4(\mathbf{p}) \cup N_D(\mathbf{p}).$$

If $\mathbf{q} \in N_8(\mathbf{p})$ then it is said that $\mathbf{q}$ is *8-adjacent to* $\mathbf{p}$. 8-adjacency is also commutative.

Other kinds of adjacencies such as mixed adjacency or *m-adjacency* for short, is somewhat more complicated, where the membership of a pixel in the neighbourhood of $\mathbf{p}$ does not only depend on its position relative to $\mathbf{p}$, but also on some values of other neighbours of $\mathbf{p}$. Consider for a start binary images only. This will be discussed later.

1.3 Processing of an image

In all image transformations we shall refer to the *input image* (i.e the image one starts with) and the *output image* (i.e. your final result). We shall normally use A as input image and B as output image. An image transformation may then be expressed as

$$B = f(A)$$

where $f(\cdot)$ constitutes some operations performed on the pixels of A to render the pixels of B. As used in this sentence, we shall often take the liberty to say 'pixel' when we actually mean 'pixel value'.