Chapter 2 Introduction to Digital Colour

Celeste Eugenia Lombardi, photographer

Abstract

This chapter explains the logic of digitising analogue colour information.

Keywords:
Digital Colour, Sampling, Quantization, Bit

DOI 10.23738/RCASB.00502
This chapter is part of the book: Color Design & Technology - A Multidisciplinary Approach to Colour – Part 1, edited by Alice Plutino, Gabriele Simone and Alessandro Rizzi
ISBN 978-88-99513-19-1

Research Culture and Science Book Series Vol 5
ISSN 2785-115X www.rcasb.eu

1. Introduction

The Introduction section should include the background and aims of the chapter in a comprehensive manner.
Once you have focused on the specific topic of your study, you should investigate the latest and most relevant literature related to your study. Your literature review should be complete, but not overly long.

1.1 Introduction to digital colour

Human beings have developed language as a form of communication: every state and regional subdivision brings more or less substantial differences to language. A native Italian speaker can easily understand other native Italian speakers; he may have some difficulty in understanding dialectal nuances; he will not be able to understand a native German speaker (unless he has studied the language).
Machines communicate with each other in this way:

01000011 01100101 01101100 01100101 01110011 01110100 01100101

Incomprehensible to humans, but clear and immediate for all machines, whether they are programmed in Japan or America.
It is called binary language and, as the name suggests, it is a positional numeral system in base 2. We can see that only two numbers are used:

0 1

If we pay attention to the multi sockets we have in our house, we can see that the same symbols are present on the red switch. The binary system works like a switch: it's either on, or off. There is no middle ground.

#B2FFFF

This, on the other hand, is the hexadecimal alphabet and, as the name suggests, is a positional number system in base 16. The symbols used are:

0 1 2 3 4 5 6 7 8 9 A B C D E F

This language is used by machines but can be understood by some humans.
On a program like Photoshop, the hexadecimal language (also called Hexa or hex) is used to define a colour.
#B2FFFF corresponds to the colour Celeste.

Three different languages carry the same message. So how can we convert a set of analogue information into a **digital image**[1]?
The output of most of the image sensors is an analogue signal, and we cannot apply digital processing on it because we cannot store it. We cannot store it because it requires infinite memory to store a signal that can have infinite values.
So we have to convert an analogue signal into a digital signal.
To create a digital image, we need to convert continuous data into digital form. There are two steps in which it is done.

- Sampling
- Quantization

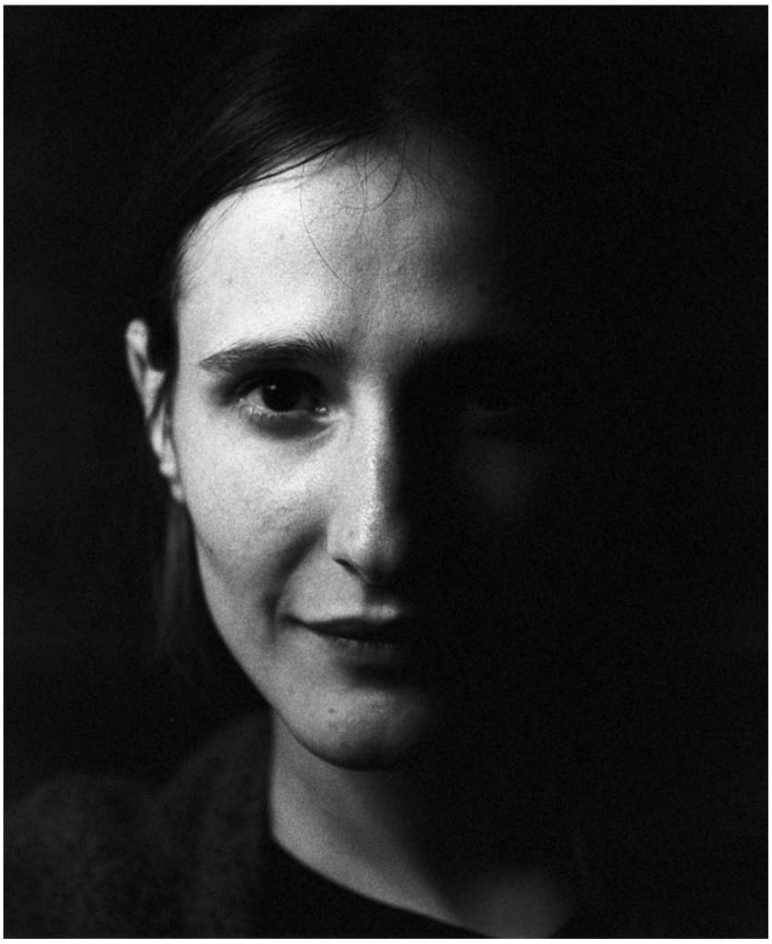
Both operations correspond to a discretization of a quantity but in different domains.

2. Sampling

Sampling corresponds to a discretization of the space (spatial resolution)
Images having higher spatial resolution are composed with a greater number of pixels[2] than those of lower spatial resolution.
The term resolution refers to the total number of counts of pixels in a digital image. For example: If an image has M rows and N columns, then its resolution can be defined as M X N.
If we define resolution as the total number of pixels, then pixel resolution can be defined with a set of two numbers: one for height and the other for width.

[1] It is a numerical representation of a two-dimensional image. Usually the image I is represented by a Matrix with dimension MxN
[2] A pixel, by definition, is the smallest element resulting from the discretization of the space. It hasn't physical dimensions and it is composed of 3 sub-pixels (one for each RGB channel): every sub-pixel has 1 value, plus the position's value.
If you open a photo with photoshop and zoom in, you will see a lot of tiny, cute, little, coloured blocks: those are the pixels!

Let's look at this image:



Taken with a Nikon f5 on Ilford HP5 Plus 400 film and digitized with a special scanner.
The spatial resolution of this image is:

3374x4121 pixel

This means that there are 3374 pixels on the short side, which we will call N from now on, and 4121 pixels on the long side, which we will call M from now on.
The total number of pixels is 13,904,254 and the weight of the JPEG file is 864 KB.

Now, let's change the spatial resolution by decreasing the number of pixels in the image:

The spatial resolution of this image is:

200x243 pixel

The total number of pixels is 48.600 and the weight of this JPEG file is 24,3 KB.

We continue in this process, further decreasing the number of pixels in the image:

The spatial resolution of this image is:

50x61 pixel

The total number of pixels is 3.050, for a weight of 13,4 KB.

Again:

The spatial resolution of this image is:

10x12 pixel

For a total of 120 pixels and a weight of 11,5 KB.
The operation we performed is called decreasing the sampling rate: we went from an image with 13 million pixels to one with only 120 pixels.
What has changed? In the last image, it is impossible to understand what is represented in the picture.
In digital images, increasing the sampling rate is equivalent to increasing the image resolution: with a higher resolution, we have more pixels (which we can also call sampling points) representing the same scene.
The weight of the image will be greater and we will have more detail.

Higher sampling rate
=
Higher image resolution
=
Higher quality of the image
=
More memory used

A natural image is coloured in continuous tones, and thus it theoretically has an infinite number of colours. The discrete and finite language of the computer restricts the reproduction of an infinite number of colours and shades. During Quantization, the infinite colour shades are synthesized into a finite list of values. This means that each colour is mapped and inserted into a single pixel.
How many colours do we want to use in an image?

This image, taken with a Nikon D750, has been quantized with 256 colours.

What happens if I lower the number of colours from 256 to 4?

Quantization reduces the number of colours used in an image: this leads to a loss of definition and fidelity because different colours are remapped into the same colour. In the example above we have two shades of blue: one darker and one lighter.
Are two colours enough to represent the infinite number of nuances that a spring sky can have? Of course not.

Low quantization rate
=
Loss of details

The quantization process is closely related to the concept of **bit depth:**

- 2-bit image will display 2^2 (4) colours
- 8-bit image will display 2^8 (256) colours.

Higher bit depth

=

More bits to represent a colour

=

More memory to use

Ok but, what is a bit?

Binary dig**IT** is a basic unit of information used in computing and digital communications. It can have only one of two values (0 / 1). Remember the binary language? Here we are.

How many numbers can we represent with 3 digits in a base-2 system?

000 - 0
001 - 1
010 - 2
011 - 3
100 - 4
101 - 5
110 - 6
111 – 7

8 numbers.

The memory of a computer usually uses an elementary unit of 8 bits: a byte

1 byte = 8 bits

The number of used bits defines the quantity of information (and the colour quality of the digital image)

8 bits = 2^8 (256) possible values.

3. Exercise time – How many bits?

Camera's megapixel: 6

Pixel resolution: 3000x2000 px

Colour depth: 24-bit

3.1 Step 1

Multiply the detectors number of horizontal pixels by the number of vertical pixels to get the total number of pixels of the detector

Horizontal: 3000 px

Vertical: 2000 px

3000 px x 2000 px = 6.000.000 px

3.2 Step 2

Multiply total number of pixels by the bit depth of the detector to get the total number of bits of data

Total number of pixels: 6.000.000 px

Bit depth: 24 bit

6.000.000 px x 24 bits/px = 144.000.000 bits

3.3 Step 3

Dividing the total number of bits by 8 equals the file size in bytes.

Total number of bits: 144.000.000 bits

144.000.000 bits / 8 bits = 18.000.000 bytes

3.4 Step 4

Divide the number of bytes by 1024 to get the file size in kilobytes. Divide by 1024 again and get the file size in megabytes.

18.000.000 bytes / 1024 = 17.578,125 Kilobyte

17.578, 125 KB / 1024 = 17,166 Megabyte

4. In summary

of pixels x Bit Depth / 8 / 1024 / 1024 = File Size in Megabytes (MB)

5. Conflict of interest declaration

The authors declare no conflict of interest.

6. Funding source declaration

This study did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

7. Short biography of the author(s)

Celeste Eugenia Lombardi, tutor at Polidesign in Milan in the Master in Colour Design and Technology, is a photographer and videographer specialised in colour as a storytelling element in audiovisual products. Since 2019, she has been collaborating with Prof. Rizzi, giving lectures for the Multimedia Project course at Università Statale di Milano, Computer Science Department.