Digital image processing and analysis

4. Colour: Digital representations

Professor Ela Claridge
School of Computer Science

Previous lecture:

- Colours and their origins
  - Physical underpinnings
  - Human visual perception
- Colour images
  - Image acquisition
  - Colour spaces

In this lecture we shall find out about:

- Digital representation of colour images
  - Colour mixing (vector arithmetics)
  - Pixel arrays
  - Colour models

Colour mixing
RGB

Mixing in RGB space is **additive**

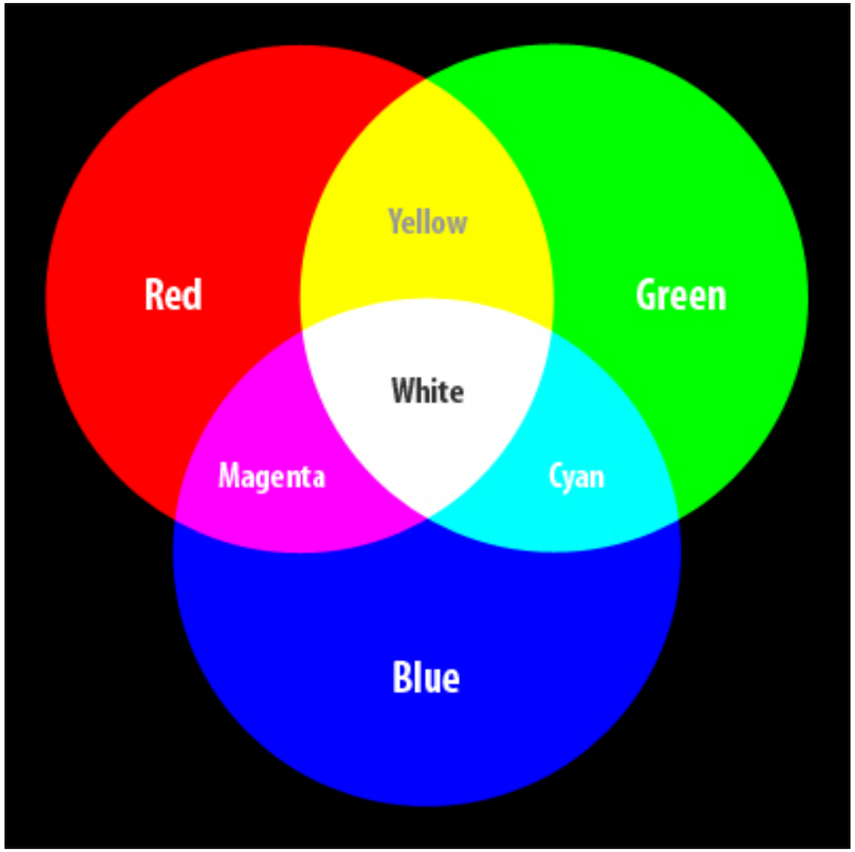


Image source: http://www.netsourceinc.com/blog/quick-color-guide

Colour mixing
RGB

Vector notation for colours

[Primary1   Primary2   Primary 3]
[R   G   B]

Examples

red = [255 0 0]
green = [0 255 0]
blue = [0 0 255]
yellow = red + green = [255 255 0]
magenta = red + blue = [255 0 255]
cyan = green + blue = [0 255 255]

orange = [255 127 0]

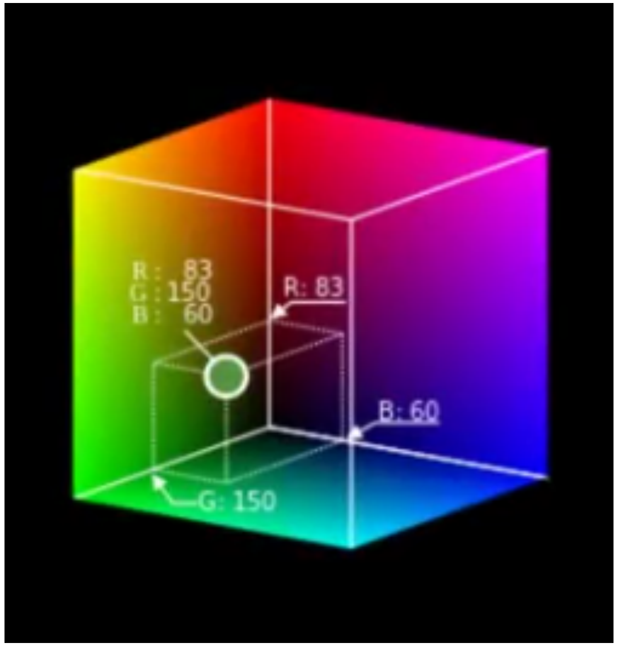
See also: https://www.youtube.com/watch?v=RY8XcwVlwgE

Colour mixing

Colour picker experiment

Colour mixing
RGB

Given that each primary is represented by 1 byte (8 bits), how many colours can be represented in total by three primaries?



$2^{3x8} = 2^{24} = 16{,}777{,}216$

Colour mixing
CMY

Mixing in CMY space is **subtractive**

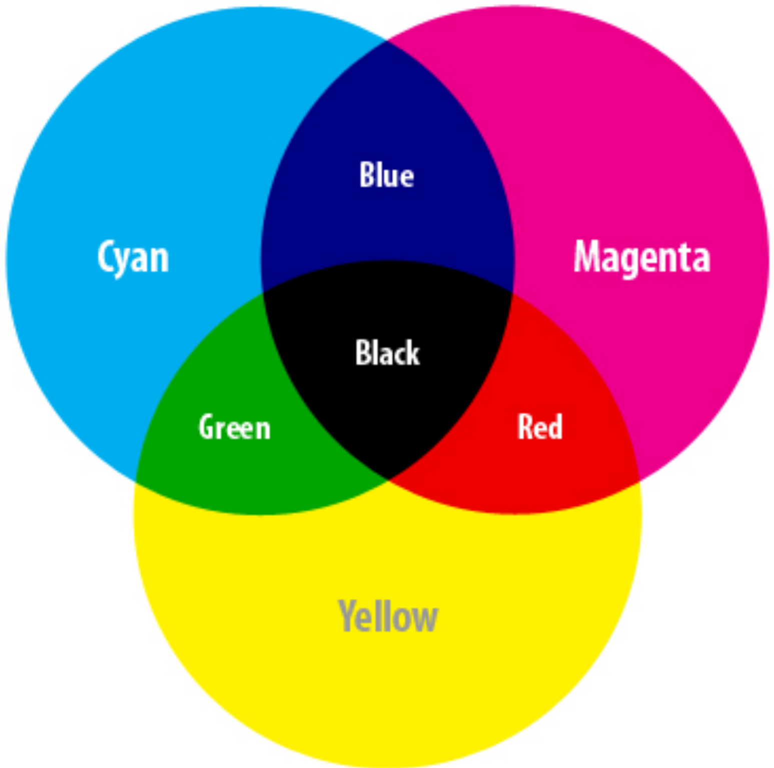


Image source: http://www.netsourceinc.com/blog/quick-color-guide

Colour mixing
CMY

[C M Y]

cyan = [255 0 0] $_{CMY}$
magenta = [0 255 0] $_{CMY}$
yellow = [0 0 255] $_{CMY}$

Example: yellow surface

Given white light

[255 255 255] $_{RGB}$

yellow pigment absorbs blue component of the spectrum

[255 255 255] $_{RGB}$ – [0 0 255] $_{RGB}$ = [255 255 0] $_{RGB}$

so a mixture of red and green is reflected

[255 255 0] $_{RGB}$ = yellow

Colour space conversion

Colour space conversion

- Colours can be converted from one space to another

- Conversion from RGB to CMY:
  $[\ C\ \ M\ \ Y\ ] = [\ 255\ \ 255\ \ 255\ ] - [\ R\ \ G\ \ B\ ]$

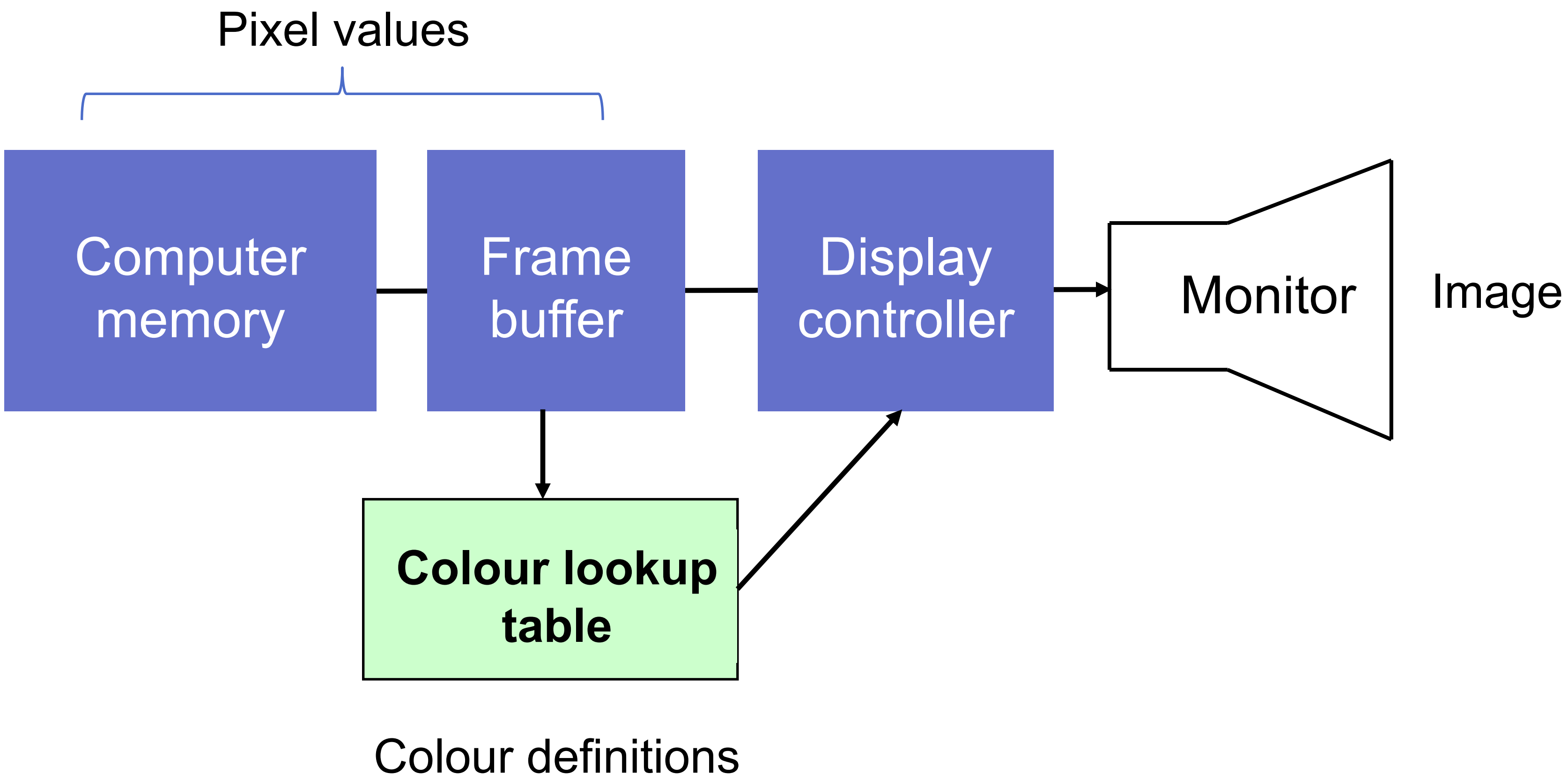
- Example: Convert green from RGB to CMY
  $[\ C\ \ M\ \ Y\ ] =$
  $[255\ \ 255\ \ 255\ ]_{RGB} - [\ 0\ \ 255\ \ 0\ ]_{RGB} =$
  $[255\ \ 0\ \ 255\ ]_{CMY}$

Colour image display

- Digital colour images have two components:
  - **raster data** - an array of pixels;
  - **colour model** - a description of how pixels are mapped to colours.

Colour image display

Computer architecture for colour displays



Colour image display
Frame buffer

- Frame (display) buffer
  - A specially designated area of memory
  - Direct access by a display processor (but not by an application)
  - Display processor scans the display buffer and passes the contents to a DAC (Digital-to-Analogue Converter)
  - DAC converts values into voltages for individual R, G and B pixel cells
- The colour lookup table is a block of fast RAM (Random Access Memory)

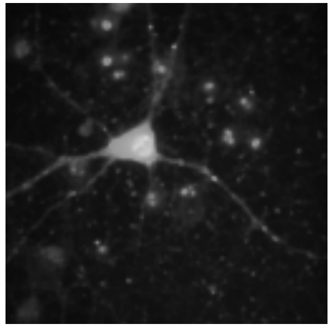
Colour image display
Raster data and pixel structure

- Raster data - raster array - a rectangular array of picture elements (pixels)
- Raster array forms a picture
- The structure of a pixel depends on
    - the colour space
    - the colour model

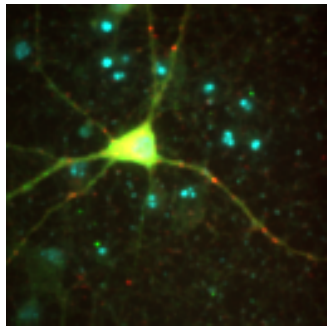
Colour image display
Raster data and pixel structure

- Binary images (purely black-and-white)
  - pixel represented by a single bit

- Monochrome images
  - pixel represented by a single byte
    (a one-dimensional vector)



- Colour images
  - usually represented by a 3-dimensional vector in a given colour space
  - examples [R G B], [H S V] or [C M Y]



Colour image display
Direct and indexed colour

- **Direct colour**: each pixel encodes its own colour information.
- **Indexed colour**: each pixel does not store colour information, but rather an index into a colour lookup table (LUT). The LUT can contain any colours.

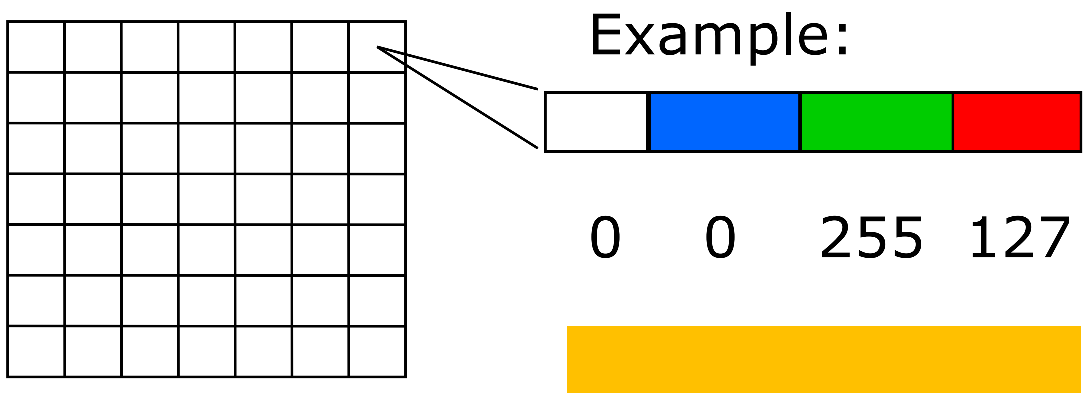
Colour models

A colour model describes how pixels are mapped into colours.

Colour models
Direct colour (Packed Array)

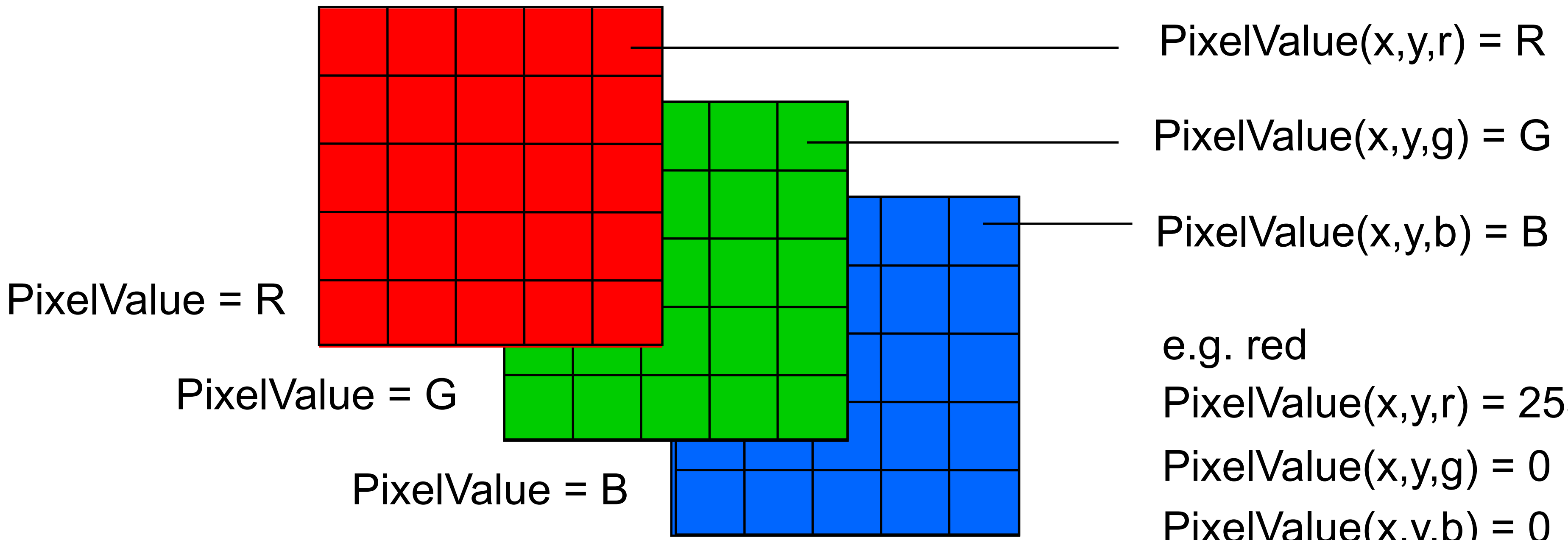
- Image is an array of **values**, each encoding a colour
    - 4-byte integer
      aaaaaaaa bbbbbbbb gggggggg rrrrrrrr

Alpha-channel

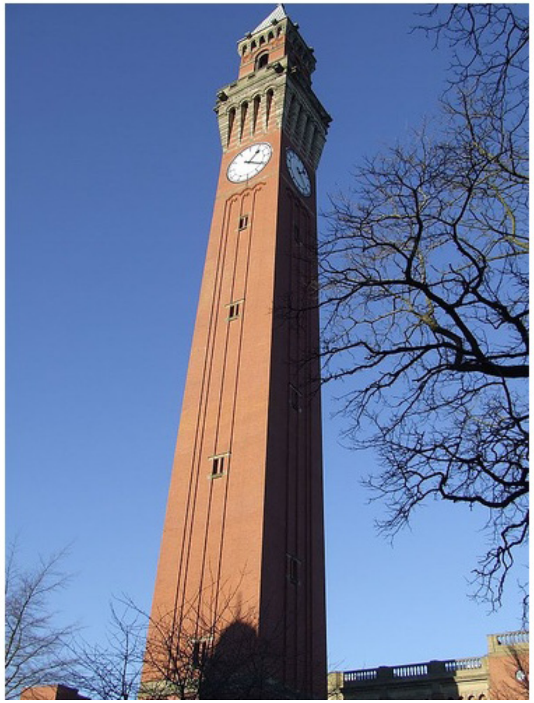


Colour models
Direct colour (True Colour)

- Image is represented by three **channels**
  - three pixel arrays, one for each primary colour
- Each channel directly encodes values of one of the three primaries



e.g. red
PixelValue(x,y,r) = 255
PixelValue(x,y,g) = 0
PixelValue(x,y,b) = 0




Red plane
Red channel


Green plane
Green channel

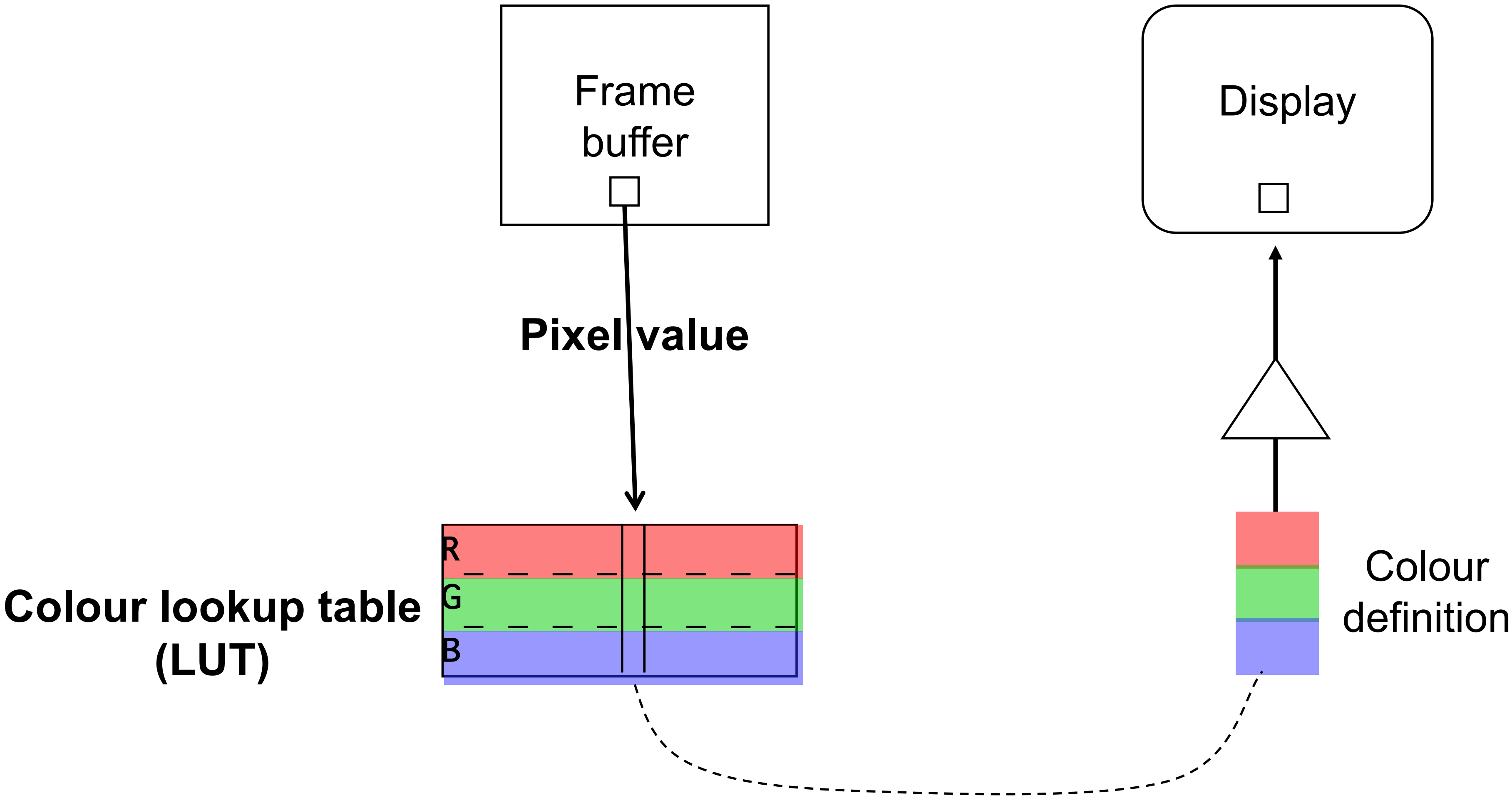

Blue plane
Blue channel

Colour models
Indexed colour

- A pixel value (or a value of a pixel component) is an index (a pointer) to a table containing colour definitions

Colour models
Indexed colour

Mapping of pixel values to colours



Colour models
Indexed colour

Colour lookup table

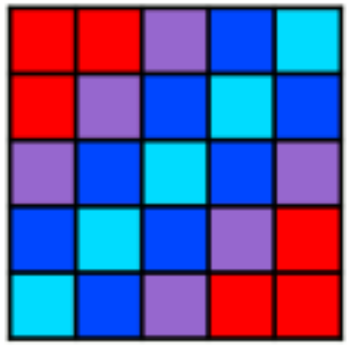
- Synonyms:
  - Colour Lookup Table
  - CLUT
  - LUT

- Each location in a LUT stores a colour definition for a pixel with a given value

0	0	1	2	3
0	1	2	3	2
1	2	3	2	1
2	3	2	1	0
3	2	1	0	0

Raster array

LUT

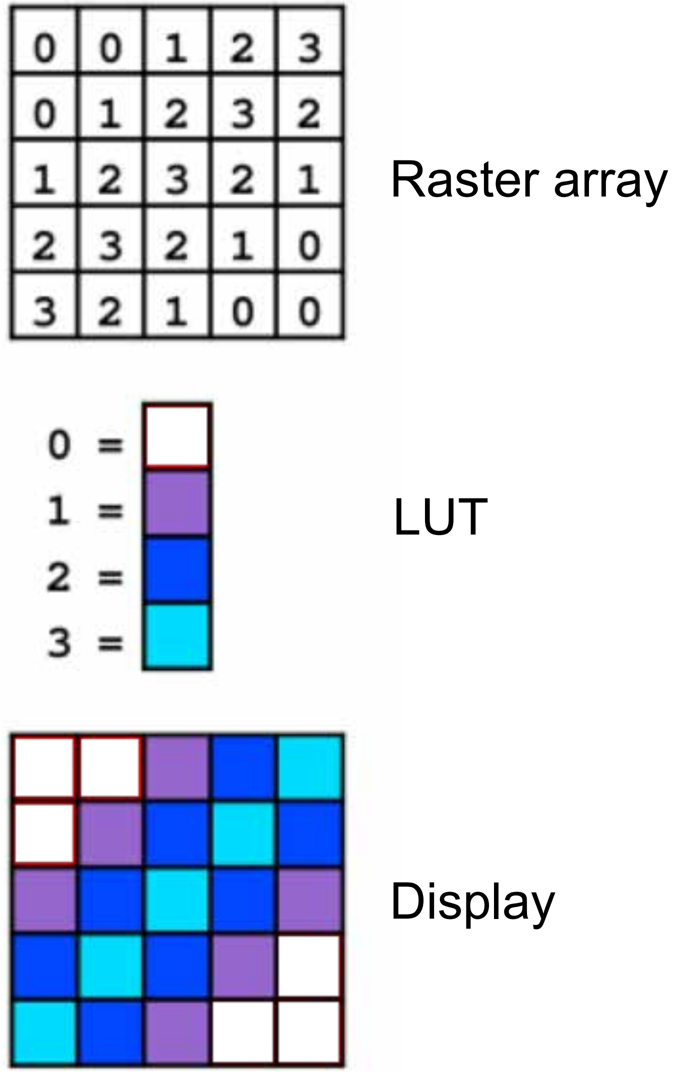


Display

Source: Wikipedia

Colour models
Indexed colour

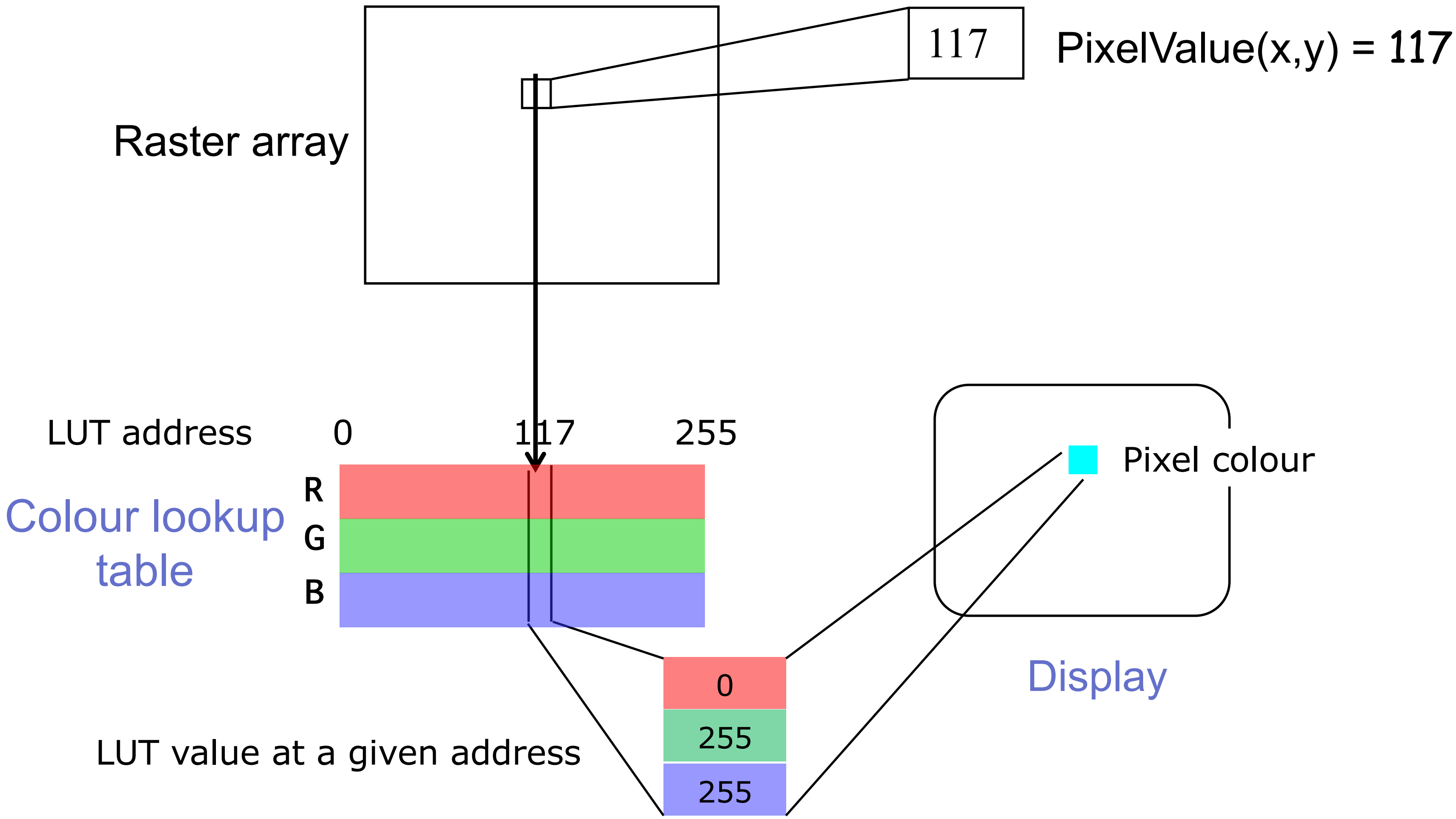
- Changed definition for colour "0" (from red to white)



Source: Wikipedia

Colour models
Indexed colour

Colour mapping for 1-byte pixels (8 bit)

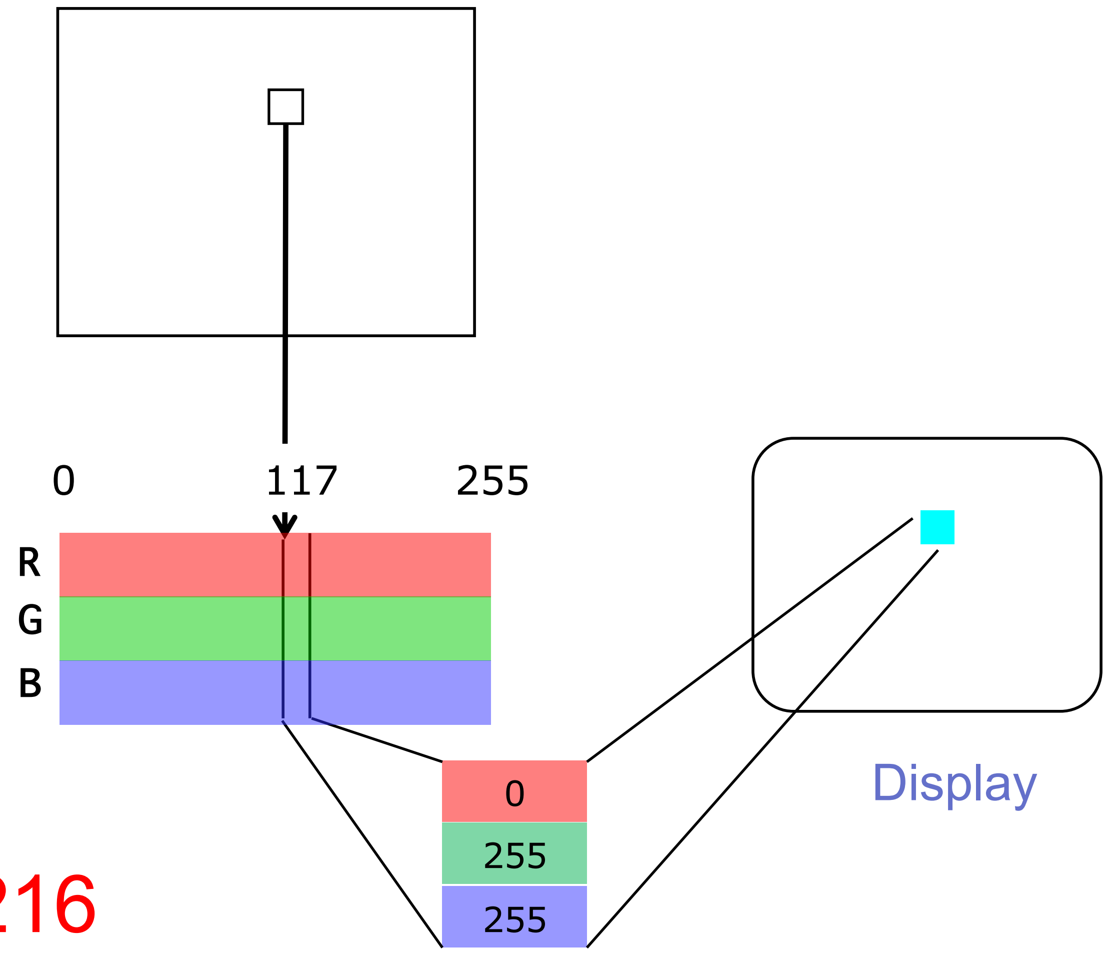


Colour models
Indexed colour

Colour mapping for 1-byte pixels (8 bit)

Number of colours **simultaneously** available in one image?

$2^8 = 256$

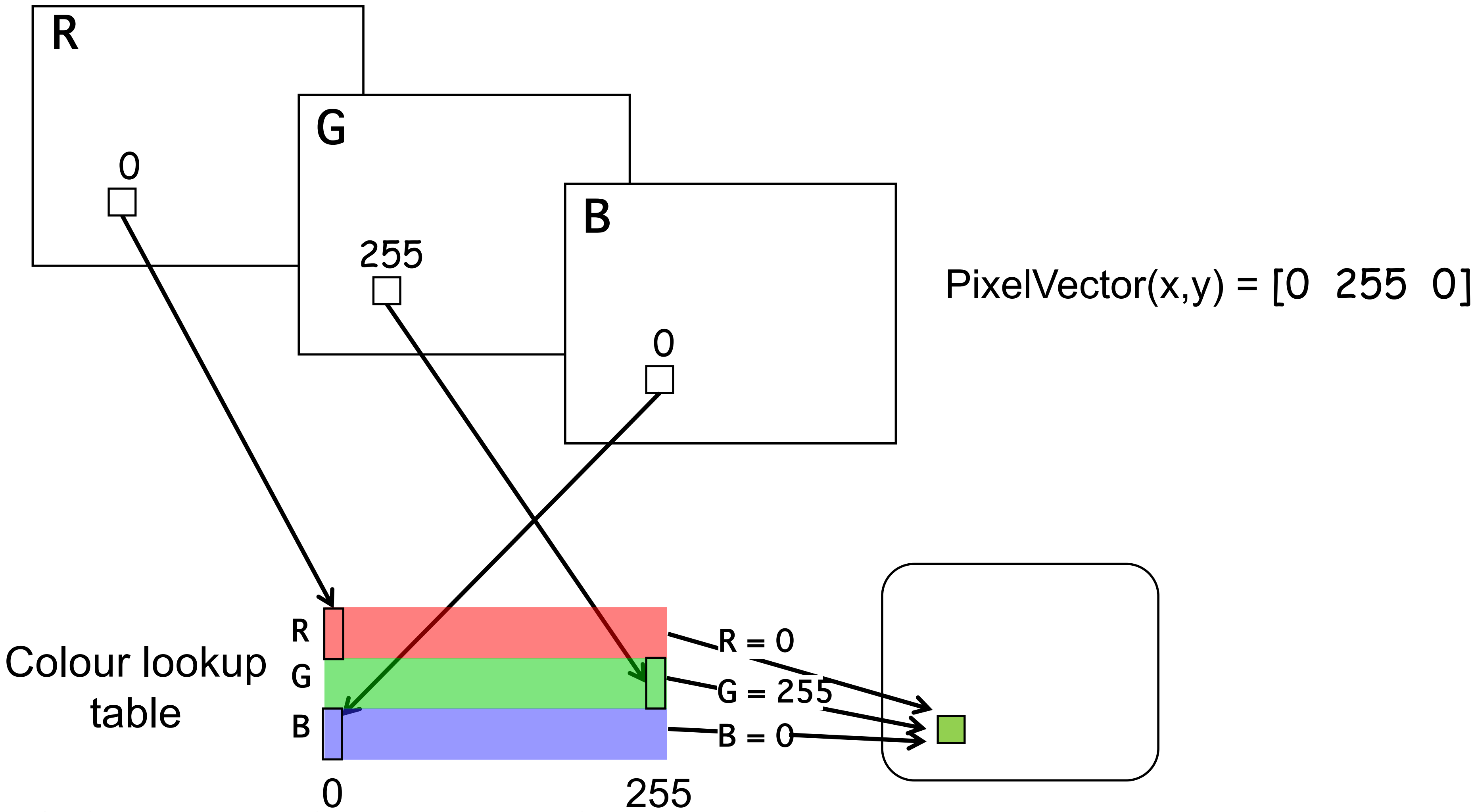


Gamut (total number of colours available for use)?

$2^{3x8} = 2^{24} = 16,777,216$

Colour models
Indexed colour

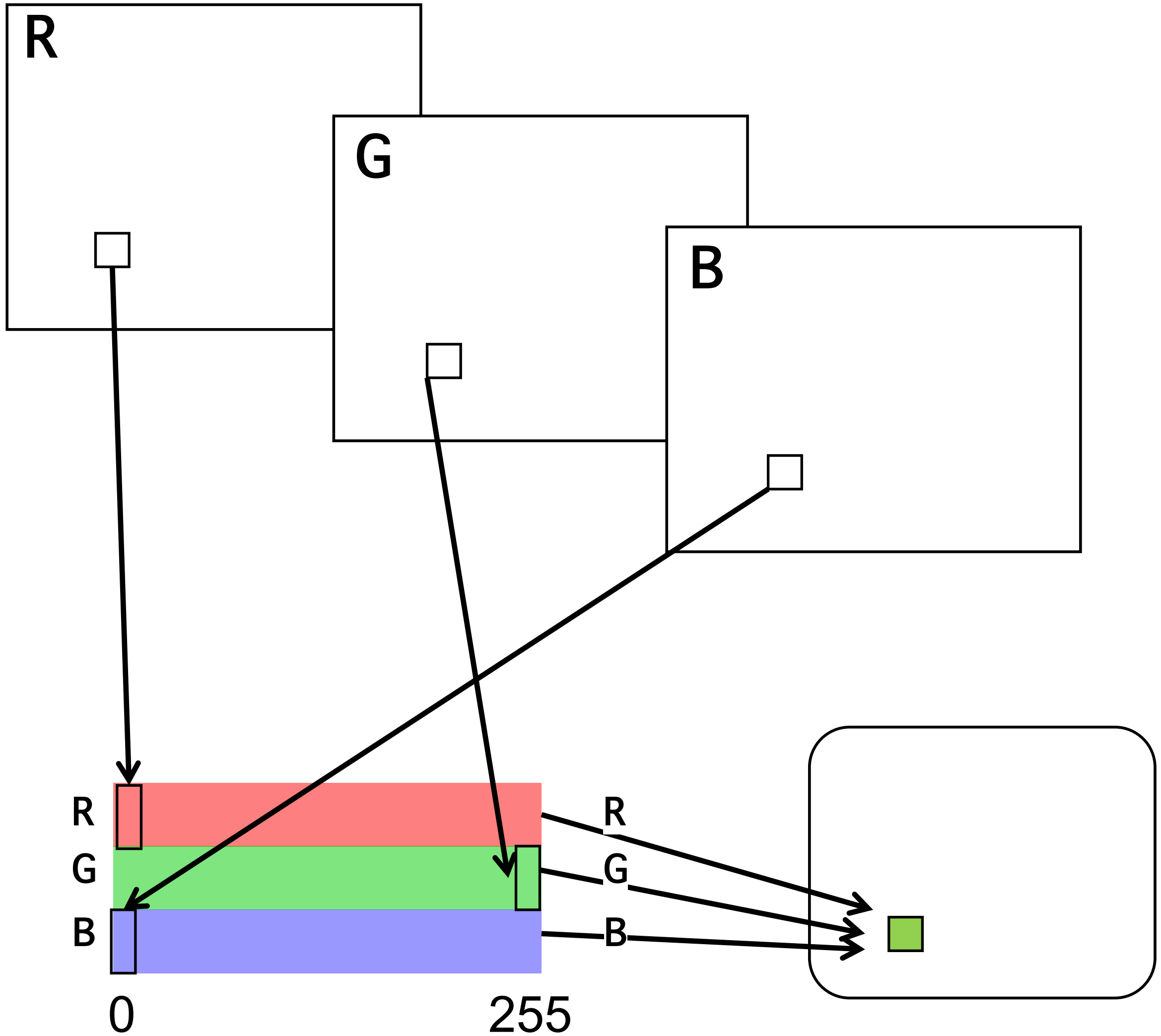
Colour mapping for 3-byte pixel vectors (24 bit)



Colour models
Indexed colour

Number of colours **simultaneously** available in one image?

$2^{24} = 16,777,216$



Gamut (total number of colours available for use)?

$2^{24} = 16,777,216$

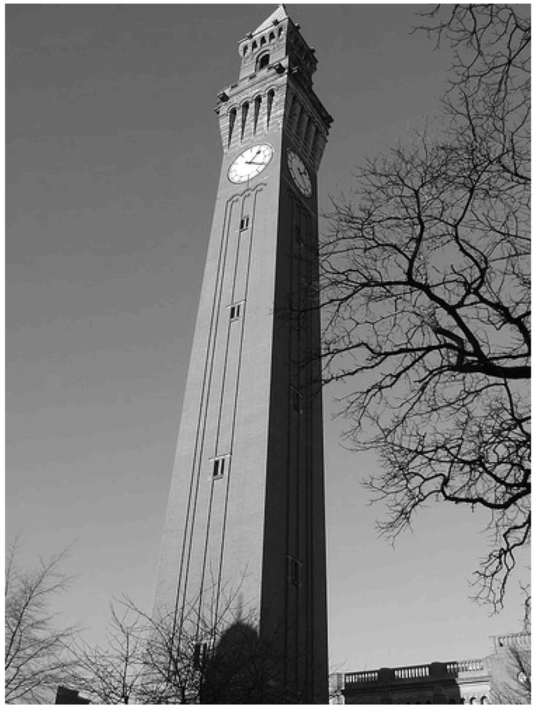
Colour models
Indexed colour

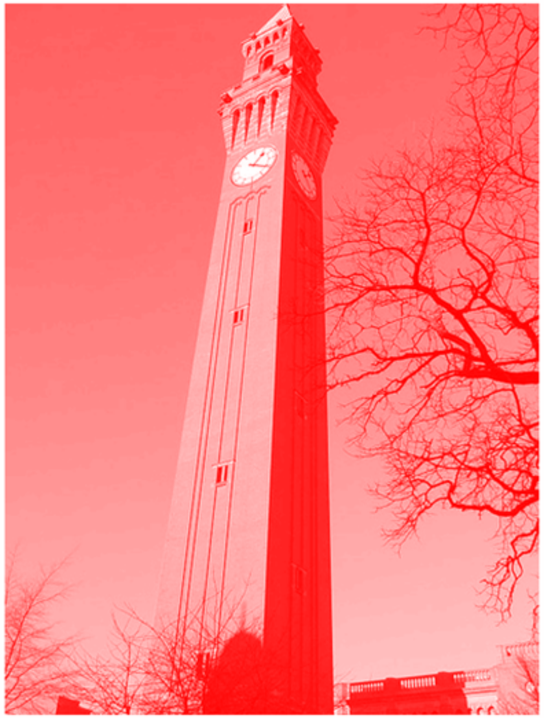
Changing pixel colours

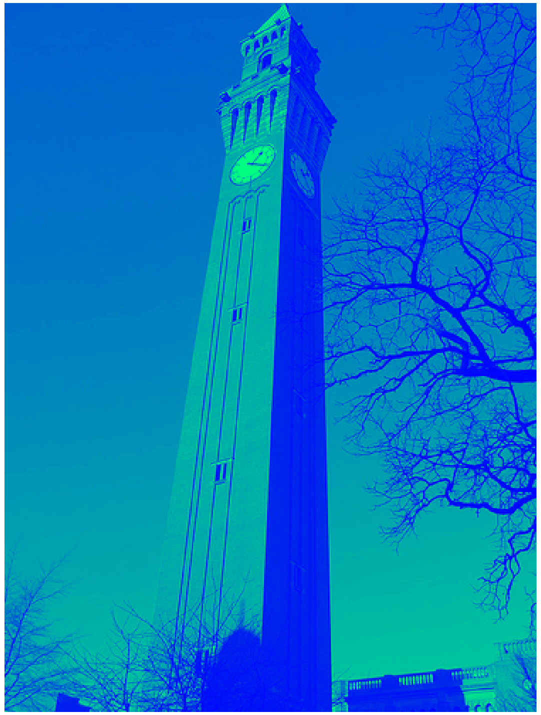
- Changing pixel colours is very easy within the Indexed Colour Model
- A raster array containing pixel values (or pixel vectors) stays unchanged.
- Only colour definitions in the LUT are changing

Colour mapping functions

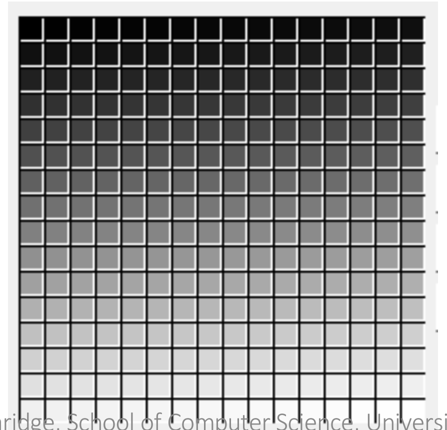
Image

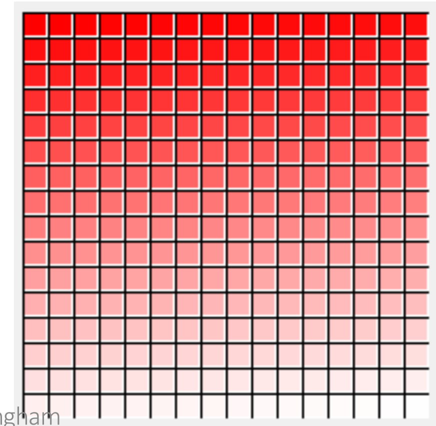


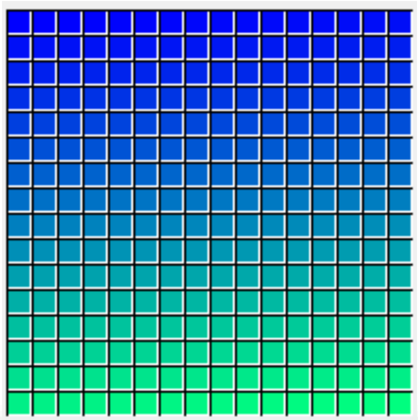




LUT
(256
positions)

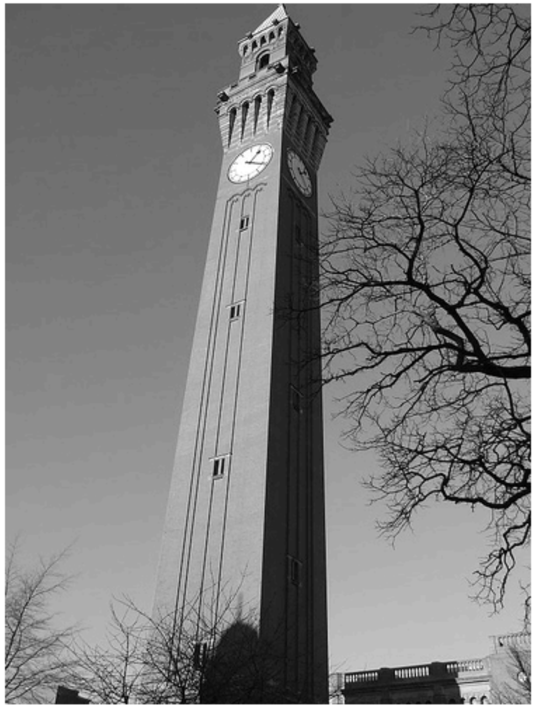


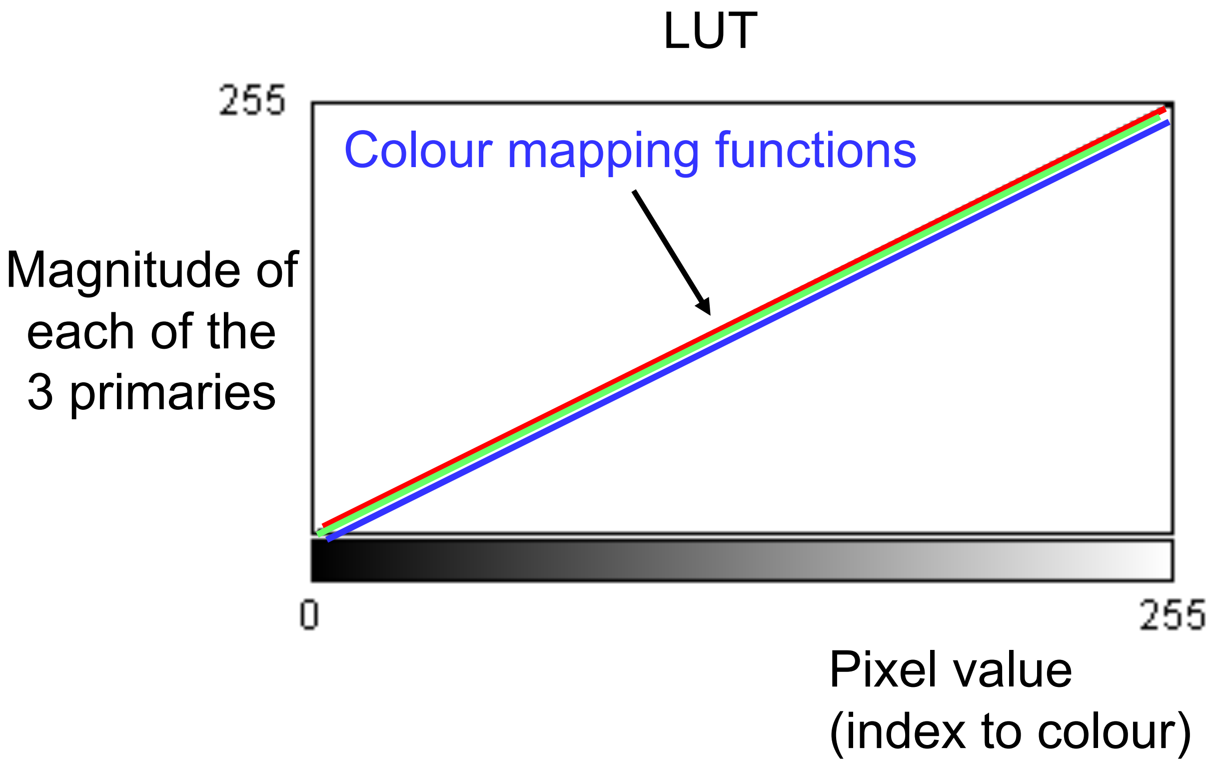




Colour mapping functions

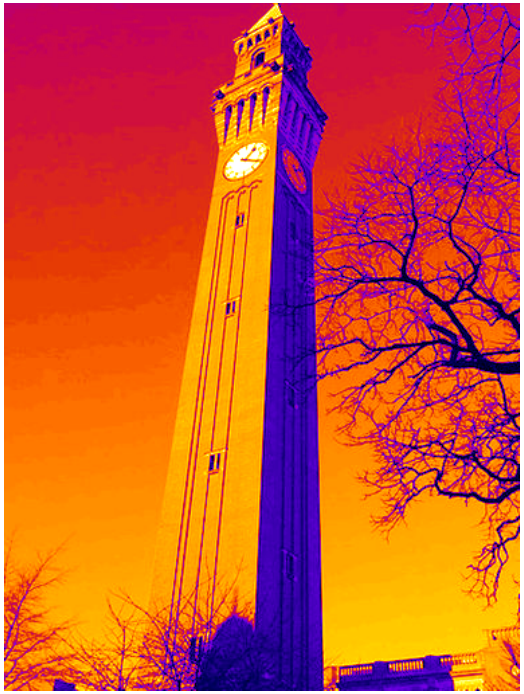
Image



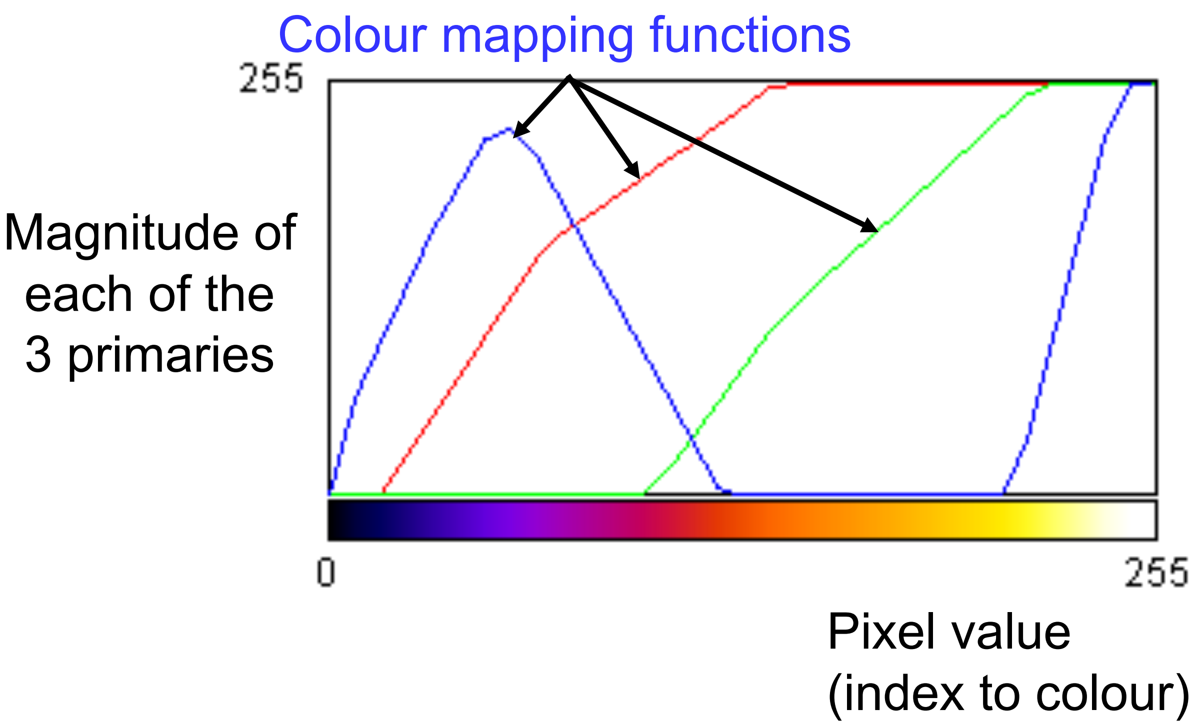


Colour mapping functions

Image

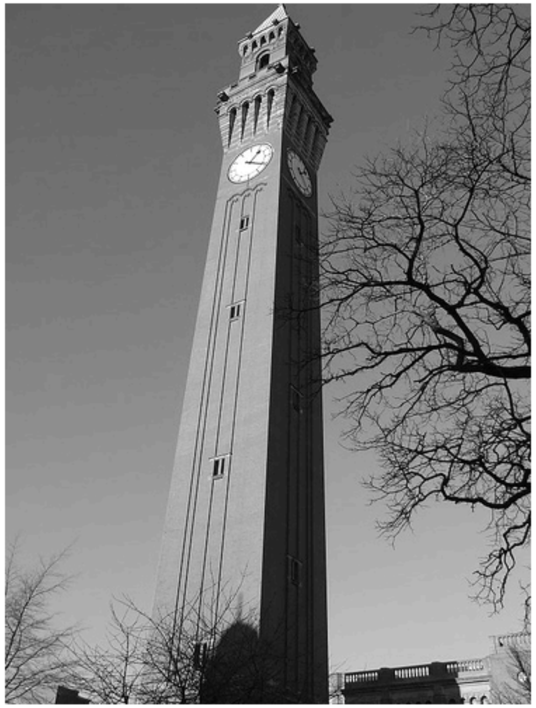


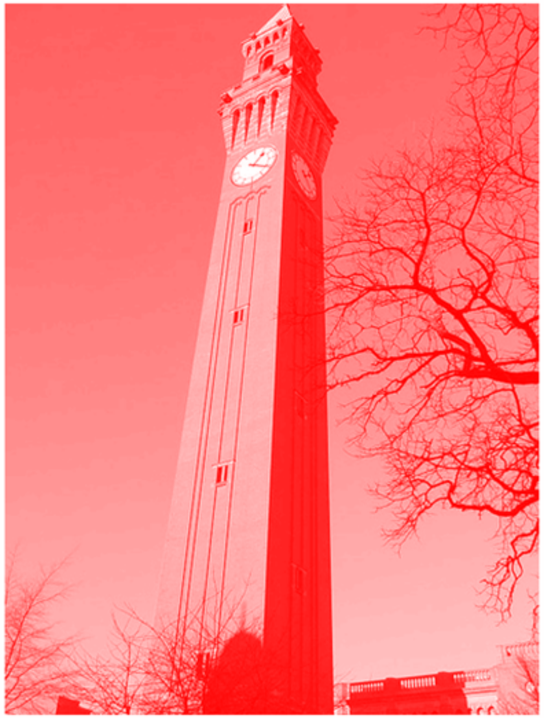
LUT

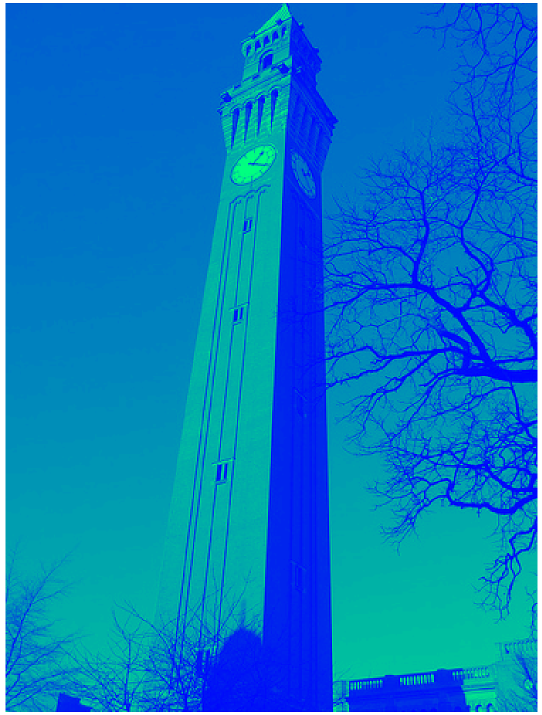


Colour mapping functions

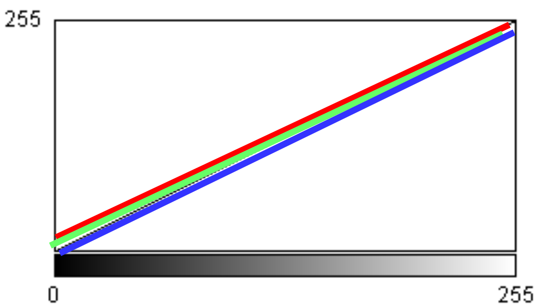
Image

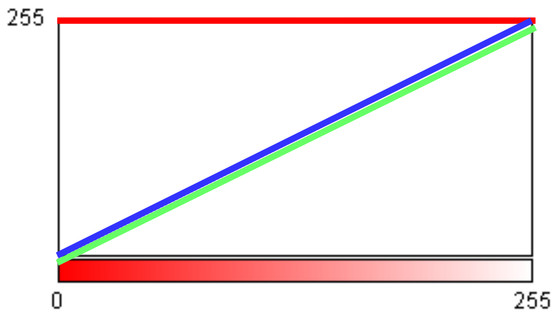


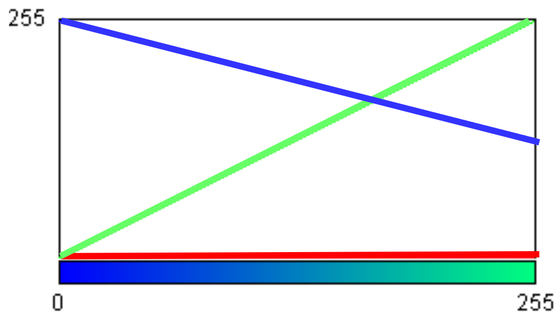




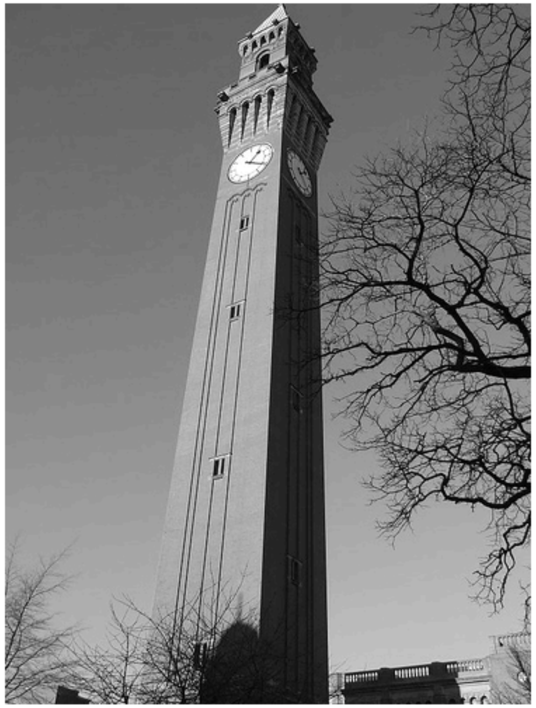
LUT

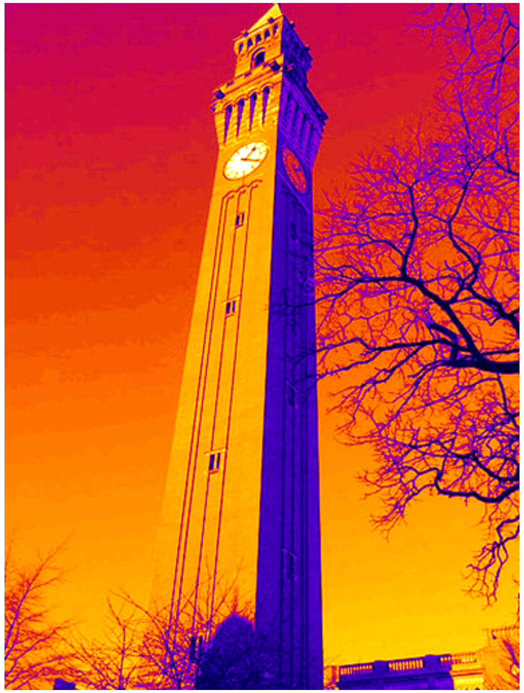


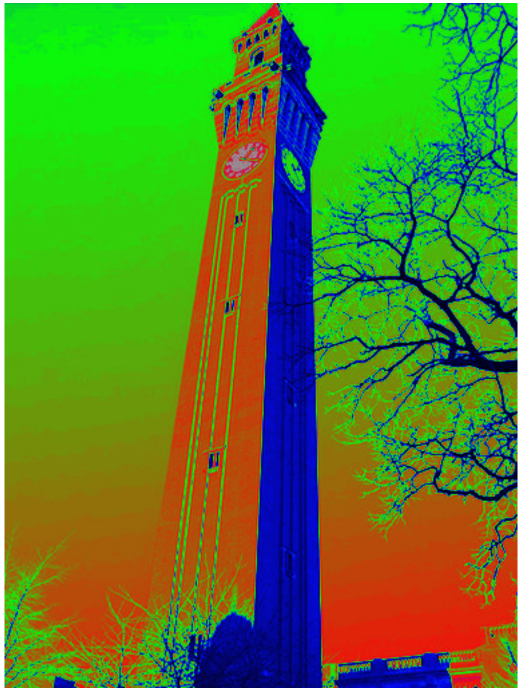


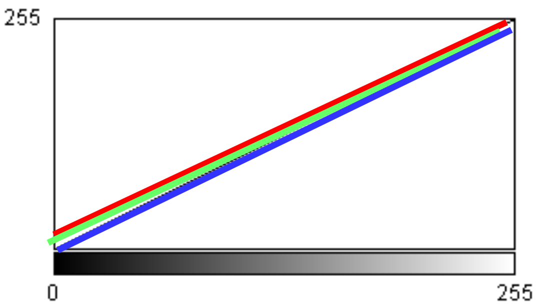


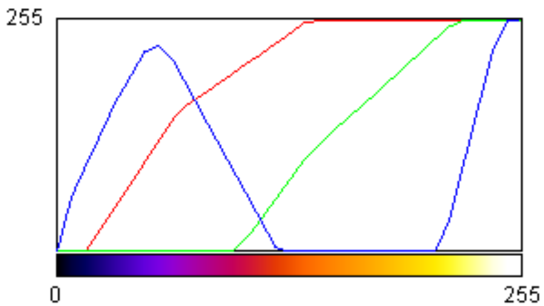
Colour mapping functions

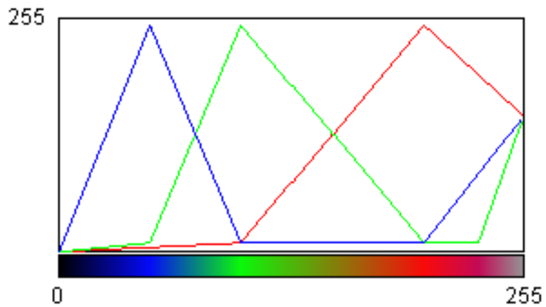












In this lecture we have covered:

- Digital representation of colour images
  - Colour mixing (vector arithmetics)
  - Pixel arrays
  - Colour models

Next lecture

- Improving image quality
  - Manipulating image brightness
  - Contrast enhancement
  - Image histogram
  - LUT operations

Further reading and experimentation

RGB colour space

- https://en.wikipedia.org/wiki/RGB_color_space

RGB to HSV conversion

- http://www.rapidtables.com/convert/color/rgb-to-hsv.htm

Basic vector and matrix arithmetic

- http://www.varsitytutors.com/hotmath/hotmath_help/topics/adding-and-subtracting-vectors
- https://mathinsight.org/matrix_vector_multiplication

Digital colour concepts

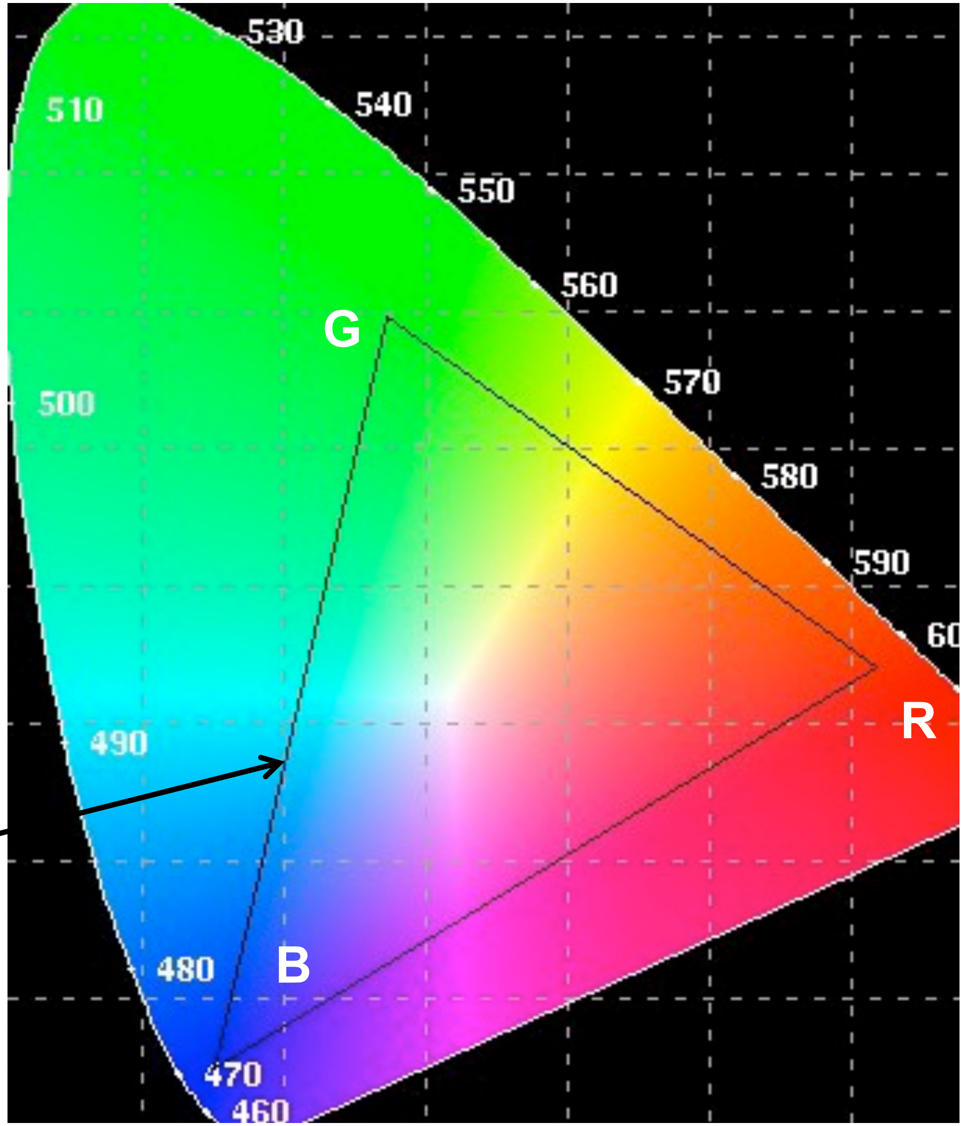
- https://en.wikipedia.org/wiki/Colour_look-up_table
- https://en.wikipedia.org/wiki/Indexed_color
- https://en.wikipedia.org/wiki/Color_depth

You may be interested to know …

Colour spaces (previous lecture)
CIE XYZ

Chromacity diagram

Gamut



Colour space conversion
RGB to XYZ

- Each of the R, G and B primaries is a weighted sum of X, Y and Z primaries

- Weights (fractions of each primary) are expressed in matrix notation, e.g.

$$\begin{bmatrix} 0.41 & 0.21 & 0.02 \\ 0.36 & 0.71 & 0.12 \\ 0.18 & 0.07 & 0.95 \end{bmatrix}$$

- The matrix values are characteristic for a given graphics device

Colour space conversion
RGB to XYZ

- Conversion implemented as a matrix multiplication (ROW x COLUMN)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.584 & 0.188 & 0.179 \\ 0.311 & 0.614 & 0.075 \\ 0.047 & 0.103 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

(This is an example conversion matrix, each device will have its own specific one)

Colour space conversion
RGB to XYZ

- Conversion implemented as a matrix multiplication

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.584 & 0.188 & 0.179 \\ 0.311 & 0.614 & 0.075 \\ 0.047 & 0.103 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.584 \cdot R + & 0.188 \cdot G + & 0.179 \cdot B \\ 0.311 \cdot R + & 0.614 \cdot G + & 0.075 \cdot B \\ 0.047 \cdot R + & 0.103 \cdot G + & 0.939 \cdot B \end{bmatrix}$$