Blatt 3

1 Digital images / color spaces

a) Explain the meaning of *digital images* and their properties. Address the following terms, at least:

- pixel
- color depth
- resolution

b) Explain the concept of a *color space* and summarize *RGB, YUV, HSV, CMYK*. For what applications were these color spaces developed?

c) Explain how to convert colors of YUV, HSV, and CMYK from and to RGB. Convert the color `R=225; G=135; B=60` (in RGB color space) to HSV, CMY, and CMYK.

d) Research how an RGB color value is stored in Java. For example, use the documentation of the class `java.awt.Color`.

a) a two-dimensional image is a graphical representation of either a physical, real-world scene or the metaphorical projection of an idea through either biological or technical neurons. Similiarly to audiosignals the digital image pipeline also contains digital to analog conversion, quantization and sampling steps. Generally speaking it is a 2D map of quantized intensity-values that translate in one way or another to a visual representation. One pixel corresponds to one part of the 2D image which can consist of one or more matrix-elements. For example one pixel in an RGB image consists of 3 2D maps of intensity values for each color channel. The color-depth is measured in Bits and is the amount of bits available to encode color information. This translates to how many different shades of grey can be represented in one colour map. The resolution of an image is the total amount of pixels that are visible in the final representation.

b) A color space is a representation system where colors are represented as coordinates. Each part of the coordinate represents one component of the colour, depending on the representation Model.

- RGB: color cube with additive mixing of three base-colours: Red, Green and Blue
  - each coordinate consists of different intensity levels of these colours
  - for modern CRT and TFT screens
- YUV: a bw-tv backwards-compatible representation of colors: the Y component represents the Luminance of the image at that point. a pixel’s colour is derived from the image’s luminance value at a given pixel.
- HSV: Hue, Saturation, Value is a more intuitive and user-friendly representation form. The *hue* sets the base color of a pixel, the saturation how intense it appears (0 = white, 100 = completely saturated) and the value how bright it is (0 = black)
- CMYK: Cyan, Magenta, Yellow, and Key (black) used for printing. It’s substractive colour mixing which means each additional colour introduced absorbs additional

wavelengths ⇒ opposite of RBG. Can be converted to RBG by complementing the values to their max intensity.

c)
RGB -> YUV: apply the given formula below

- Luminance Y‘
    - Y = 0.299 · R + 0.587 · G + 0.114 · B
- Chrominance U, V
    - U = B – Y
    - V = R – Y

YUV -> RGB

```
Y -= 16;
U -= 128;
V -= 128;
R = 1.164 * Y             + 1.596 * V;
G = 1.164 * Y - 0.392 * U - 0.813 * V;
B = 1.164 * Y + 2.017 * U;
```

RGB -> HSV

- divide RGB by 255 to map to [0;1] R’G’B’ (normalization)
- extract C_{max} = max(R,G,B) and C_{min} = min(R,G,B)
- $\Delta = C_{max} - C_{min}$
- calculate hue
    - $$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$
- calculate saturation
    - $$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$
- calculate value
    - C_{max}

HSV -> RGB

- normalize values for V and S
- calculate Chroma:
    - C = V * S
- intermediary value x based on hue
    - X = C × (1 – |(H/60) % 2 – 1|)
- calculate match value
    - M = V - C
- calculate RGB component based on Hue

- ***For*** $0 \le H < 60$: $(R, G, B) = (C, X, 0)$
- ***For*** $60 \le H < 120$: $(R, G, B) = (X, C, 0)$
- ***For*** $120 \le H < 180$: $(R, G, B) = (0, C, X)$
- ***For*** $180 \le H < 240$: $(R, G, B) = (0, X, C)$
- ***For*** $240 \le H < 300$: $(R, G, B) = (X, 0, C)$
- ***For*** $300 \le H < 360$: $(R, G, B) = (C, 0, X)$
- ***Example:*** *Since* $H = 220$, $R = 0$, $G = 0.19$, $B = 0.56$
  - convert RGB values to final values by adding Match value
  - R = R+M etc.
  - convert to 255 if necessary

RGB -> CMYK // CMYK -> RGB

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

- $K \equiv \min \{C,M,Y\}$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \rightarrow \begin{bmatrix} C - K \\ M - K \\ Y - K \end{bmatrix}$$ “Undercolor removal”

$$\left.\begin{array}{l} R = 225 \xrightarrow{\text{normalize}} 0{,}88 \\ G = 135 \longrightarrow 0{,}53 \\ B = 60 \longrightarrow 0{,}24 \end{array}\right\} C_{max} = R,\ \Delta = 0{,}65$$

$\Rightarrow 60° \cdot \frac{G-B}{\Delta} \bmod 6 =$

$= 60° \cdot \frac{0{,}29}{0{,}65} \bmod 6 =$

$H = \underline{27{,}\overline{27}°}\ \left(\frac{300}{11}\right)$

$S = \frac{\Delta}{C_{max}} = \frac{0{,}65}{0{,}88} = \underline{0{,}73}\ \left(\frac{11}{15}\right)$

$V = C_{max} = \underline{0{,}88}$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} 225 \\ 135 \\ 60 \end{bmatrix} = \begin{bmatrix} 30 \\ 120 \\ 195 \end{bmatrix}$$

$C = 30$
$M = 120$
$Y = 195$

$K = \min (CMY) = 30$

$C = 0$
$M = 90$
$Y = 165$
$K = 30$

d)

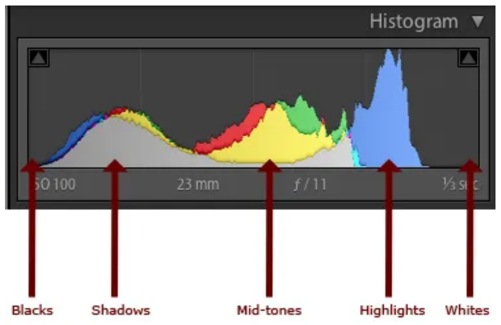
implicitly each color(channel) is completely opaque (“1” or “255”), and it’s encoding the standard RGB colorspace (sRGB). The given class can convert from RGB to HSB (HSV)

2 Histograms of digital grayscale images

Explain the concept of a *histogram* in the field of digital image processing. In particular, address the following questions:

- What information can be obtained from a histogram?
- What information cannot be obtained from a histogram?
- Which properties of a digital image can be changed by histogram manipulation?
- Can you use a histogram manipulation to subsequently erase errors made during image capture?

each bar in the histogram represents the amount of pixels of a given colour at a given intensity. To many pixels on the very far left or right side represent an under- or overexposure.

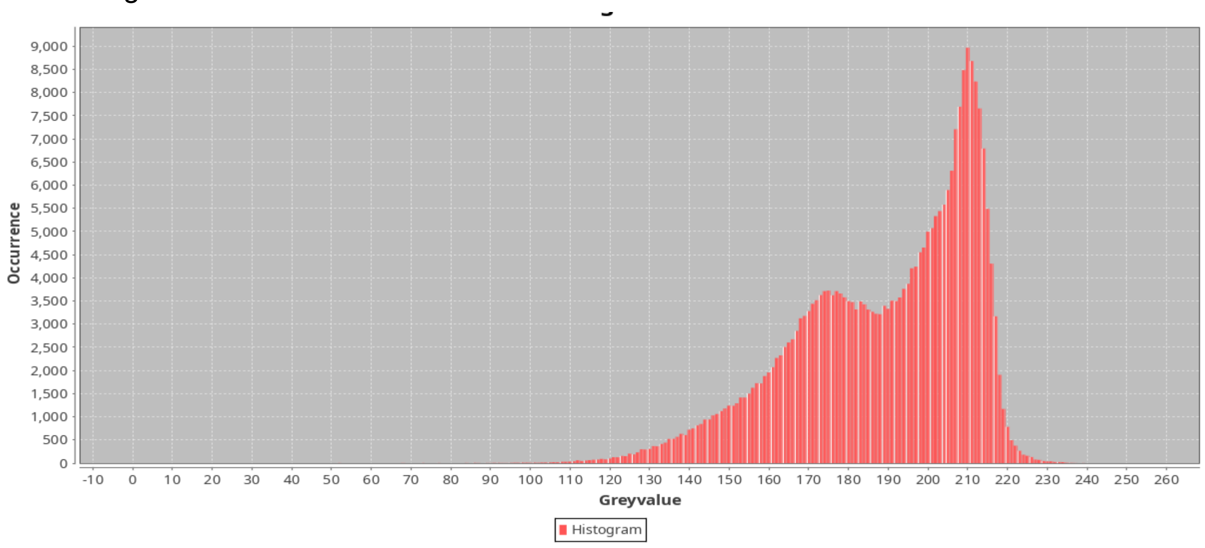


no spatial information can be obtained about composition or where over/underexposure happened. Information about texture, about objects in the scene, noise levels and type, light source information, perspective and angle, back/foreground distinction, sharpness information, resolution, compression artifacts

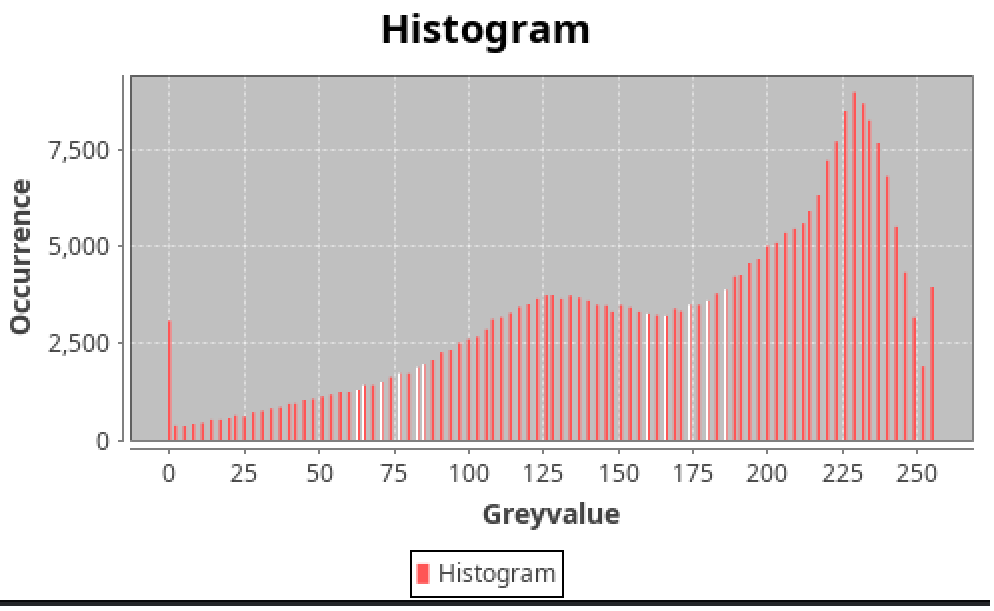
You can equalize a histogram in order to improve its contrast ratio as the pixels in the image are distributed over the available brightness spectrum

no, it cant. you can only modify brightness, contrast and tonal distribution. So no lightning errors, no focus-change, no composition change …

First histogram:



stretch to



3.)

Color Look-up Table (CLT) Principles:

- Map Colors: Transforms input colors to target colors.
- Pre-calculated Colors: Stores new color values for efficiency.
- Quick Adjustments: Speeds up color processing.

- **Image Compression:** reducing the number of colors and storing indices instead of full color values -> save storage space and bandwidth.
- **Color Quantization:** enable image representation with a limited color palette.
- **Color Correction and Effects:** they can be used to apply color transformations, corrections, or special effects by modifying the mapping of input colors to output colors.

reconstruct:

- mask value to unsigned integer (0–255)
- expand back to 8 bits, use bitmask to isolate one color from the others
- scale from reduced bits to 8 bits again