# Assignment #1

Johnathan Spinelli (spinellj)

January 21, 2022

## 1 Theory

### 1.1 Phone Camera Parameters

#### 1.1.1

Camera resolution is used to describe the size of a digital image, usually measured in megapixels [1]. Resolution can be calculated by multiplying the vertical and horizontal pixel dimensions together. An image with a higher resolution has better quality, and means the image can be made larger while retaining its detail in comparison to an image with a lower resolution [1].

#### 1.1.2

Camera sensors have pixels which convert light from photons to photoelectrons [2]. These sensors are very small, requiring the pixels to be micrometres in size. Pixel size refers to the physical size of each pixel on the sensor. Larger pixels are able to collect more photons, which results in a more sensitive sensor [2]. Although smaller pixels allow for more pixels to fit on the sensor, providing a higher spatial resolution, their inability to capture less photons reduces the sensitivity of the sensor [2].

#### 1.1.3

A common autofocus method is PDAF, which stands for Phase Detection Autofocus [3]. PDAF is often used because it is faster and more accurate than other methods of autofocusing [3]. Light shines through the camera lens and converges to a meeting point, which should be located on the sensor. If the lens is not properly focused, the meeting point will either be behind or in front of the sensor. This leads to blurry image, as the light would be striking the sensor at two separate locations instead of one. To solve this, some cameras have dedicated phase detecting photodiodes throughout the image sensor to check if the light is properly striking the sensor at one location [3]. If not, the camera knows to refocus the camera until these dedicated photodiodes receive the correct light [3].

#### 1.1.4

Shutter speed refers to the speed that the shutter closes, which directly relates to the duration of exposure [4]. As the shutter is open, light enters the camera and strikes the sensor, providing the camera with an image [4]. As the shutter remains open, more light will enter, which will affect the resulting image. When using a high shutter speed, the shutter is open for a short duration of time, which results in an instantaneous snapshot of the environment, which can be very useful for moving objects [4]. However, a short exposure means that less light will reach the sensor, often leading to a darker image[4]. When taking a picture of a darker environment, a lower shutter speed should be used [4].

#### 1.1.5

OIS stands for Optical Image Stabilization, which is used to reduce motion blur [5]. OIS controls the positioning of the lens to make sure that light is hitting the sensor in the correct locations [5]. Movement sensors such as gyroscopes can detect the motion of the camera, which can then appropriately move the lens [5]. Slower shutter speeds are susceptible to motion blur; if the target or the camera moves while the shutter is still open, light from the same source may strike the sensor at a different location, which can ruin the image. With OIS, the lens can be moved to make the light correctly strike the original sensor, improving the image quality.

### 1.1.6

ISO sensitivity refers to a camera setting that can brighten or darken a photo [6]. ISO originally referred to the type of film used to take the picture, but now it has been adopted by digital camera manufacturers [6]. Although ISO and shutter speeds can affect the brightness of an image, ISO can lead to an increase of grain/noise in the resulting image [6]. ISO is often used in scenarios where changing the shutter speed is not an option, like where motion blur is a possibility. A high shutter speed may result in a darker image, but the ISO can brighten it up to look normal [6].

## 1.2 Gamma Correction

Gamma correction is used to control the overall brightness of an image. Some images may be too bright and look bleached out, while other images may be too dark and look shadowed [7]. Through gamma correction, the brightness can be adjusted to balance the image, potentially making it easier for the viewer to notice details that previously blended in with their surroundings. The formula for gamma correction is:

$$V_{out} = V_{in}^{\frac{1}{\gamma}}$$

This equation is assuming that the values fall within the range [0,1]. If the values fit between [0,255], the input values must first be mapped before and after the calculation.

$$V_{out} = 255 \left( \frac{V_{in}}{255} \right)^{\frac{1}{\gamma}}$$

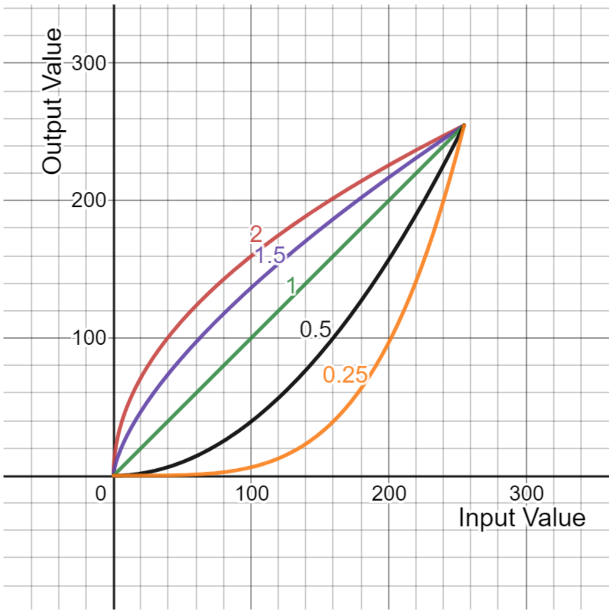


Figure 1: Gamma Coefficient Plot for $\gamma$ = 0.25,0,5,1,1,5,2

## 1.3 Other Colour Spaces

The XYZ colour space was designed to map out the colours that the average human can see [8]. The Y value corresponds to the relative luminance, while the X and Z values provide information about how the cones in the eye respond to lights with different frequencies [8]. These values are limited within a region - values beyond the region depict *imaginary* colours [8].

The xyY colour space provides a better representation of colour, as luminance in the Y is separated from the colour values, also known as the chromaticity [8]. Luminance is a measure of the intensity of emitted light (that closely corresponds to brightness) [9]. Pictured below is a chromaticity diagram, where we can see the effects of the x and y values on the resulting colour. Not pictured on the diagram is the luminance, which is how darker colours are achieved.

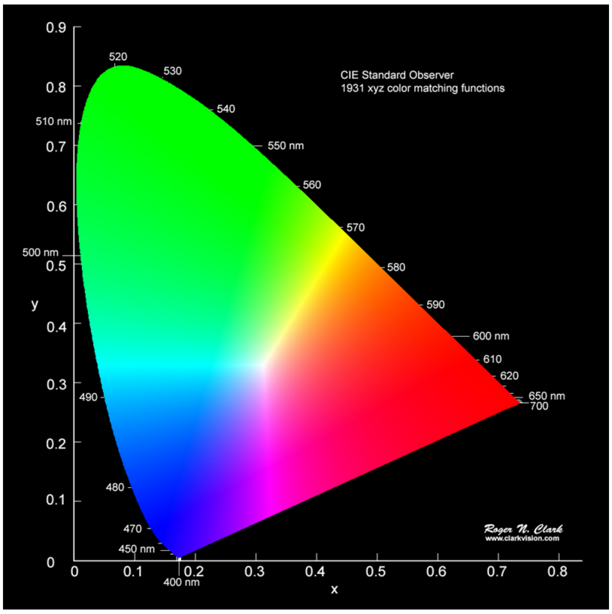


Figure 2: Chromaticity Diagram [10]

### 1.4 Interpolation

Interpolation is used to estimate the value of a point using the known values of neighbouring points. Bilinear interpolation involves interpolating in both the x and y direction, using the values and proximity of surrounding points. Bilinear interpolation can be used to upscale an image by determining values for new, intermediate pixels.

$$R1 = Q_{11} * \frac{x_2 - x}{x_2 - x_1} + Q_{21} * \frac{x - x_1}{x_2 - x_1} = 10 * \frac{50 - 20}{50 - 10} + 60 * \frac{20 - 10}{50 - 10} = 22.5$$

$$R2 = Q_{12} * \frac{x_2 - x}{x_2 - x_1} + Q_{22} * \frac{x - x_1}{x_2 - x_1} = 100 * \frac{50 - 20}{50 - 10} + 70 * \frac{20 - 10}{50 - 10} = 92.5$$

$$P = R_1 * \frac{y_2 - y}{y_2 - y_1} + R_2 * \frac{y - y_1}{y_2 - y_1} = 22.5 * \frac{40 - 30}{40 - 10} + 92.5 * \frac{30 - 10}{40 - 10} = 69.2$$

## 2 Implementation

### 2.1 Hello OpenCV

```
import cv2

orig_img = cv2.imread('img1.png', 1)

angle = int(input("Enter angle: 0, 90, 180, 270   "))

if (angle == 0):
    cv2.imshow('output',orig_img)
elif (angle == 90):
    image = cv2.rotate(orig_img, cv2.ROTATE_90_CLOCKWISE)
    cv2.imshow('output',image)
    cv2.imwrite("90_degrees.jpg",image)
elif (angle == 180):
    image = cv2.rotate(orig_img, cv2.ROTATE_180)
    cv2.imshow('output',image)
    cv2.imwrite("180_degrees.jpg",image)
else :
    image = cv2.rotate(orig_img, cv2.ROTATE_90_COUNTERCLOCKWISE)
    cv2.imshow('output',image)
    cv2.imwrite("270_degrees.jpg",image)
```

Figure 3: img1 rotated 0, 90, 180, 270 degrees

### 2.2 Gamma Correction

```
import cv2
import numpy as np

orig_img = cv2.imread('img1.png', 1)
# Hardcoded value, can prompt user for input instead
gamma = 1.5
inv = 1.0/gamma

# Method 1
table = np.array([((i / 255.0) ** inv) * 255
                  for i in np.arange(0, 256)]).astype("uint8")

out_img = cv2.LUT(orig_img, table)

# Method 2
out2 = (((orig_img / 255) ** (inv)) * 255).astype('uint8')

cv2.imshow('Original', orig_img)
cv2.imshow('Method1',out_img)
cv2.imwrite('img1_gamma.png',out_img)
cv2.imshow('Method2', out2)
```

Figure 4: img1 with gamma = 1, gamma = 1.5, gamma = 2

### 2.3 Skin Detection

```
import cv2
import time
from matplotlib import pyplot as plt
import numpy as np

'''
# Capturing webcam image
```

```
cam = cv2.VideoCapture(0)
time.sleep(1)
result, image = cam.read()
time.sleep(1)
result, image = cam.read()
cv2.imwrite('webcam.jpg',image)
'''

# Reading in the image I previously captured
image = cv2.imread('webcam.jpg', 1)
result = 1

if result:
    cv2.imshow("Picture", image)
    #cam.release()

    # Create Histogram
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    hist = cv2.calcHist([hsv],[0],None,[256],[0,256])
    plt.plot(hist, color='k')

    # Threshold acquired from examining histogram
    upper_thresh = 37
    lower = np.array([0, 0, 0])
    upper = np.array([upper_thresh,255, 255])
    mask = cv2.inRange(hsv, lower, upper)
    cv2.imshow('mask',mask)
    cv2.imwrite('mask.jpg',mask)

    # Show the resulting image, write it out
    final = cv2.bitwise_and(image,image,mask = mask)
    cv2.imshow('final',final)
    cv2.imwrite('skin_detection.jpg',final)

    # Display histogram at end, blocking menu
    plt.show()

else:
    print("something didnt work")
    cam.release()

cv2.waitKey(0)
cv2.destroyAllWindows()
```

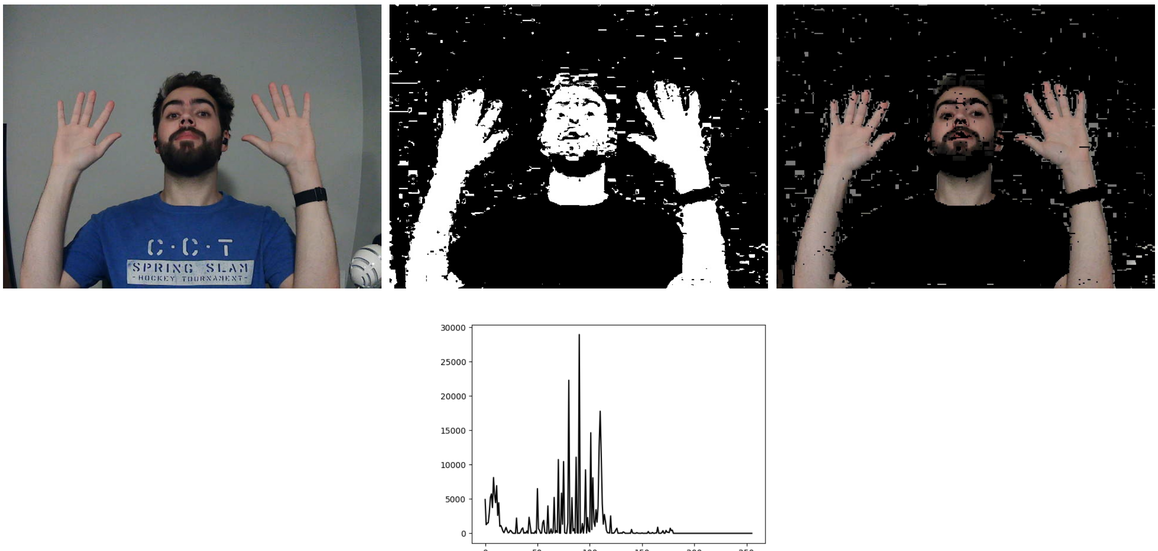


Figure 5: Original image, mask, and extracted skin. Histogram plotting hue values

## References

[1] http://facweb.cs.depaul.edu/sgrais/camera_res.htm
[2] https://www.princetoninstruments.com/learn/camera-fundamentals/pixel-size-and-camera-resolution
[3] https://www.androidauthority.com/how-pdaf-works-1102272/
[4] https://www.adobe.com/ca/creativecloud/photography/discover/shutter-speed.html#:~:text=Shutter%20speed%20is%20exactly%20what,the%20photographer%20a%20longer%20exposure.
[5] https://newatlas.com/what-is-ois-optical-image-stabilization/42212/
[6] https://photographylife.com/what-is-iso-in-photography#what-camera-iso-should-you-use
[7] https://www6.uniovi.es/hypgraph/color/gamma_correction/gamma_intro.html
[8] https://ninedegreesbelow.com/photography/xyz-rgb.html
[9] https://colorusage.arc.nasa.gov/lum_and_chrom.php
[10] https://clarkvision.com/articles/color-cie-chromaticity-and-perception/