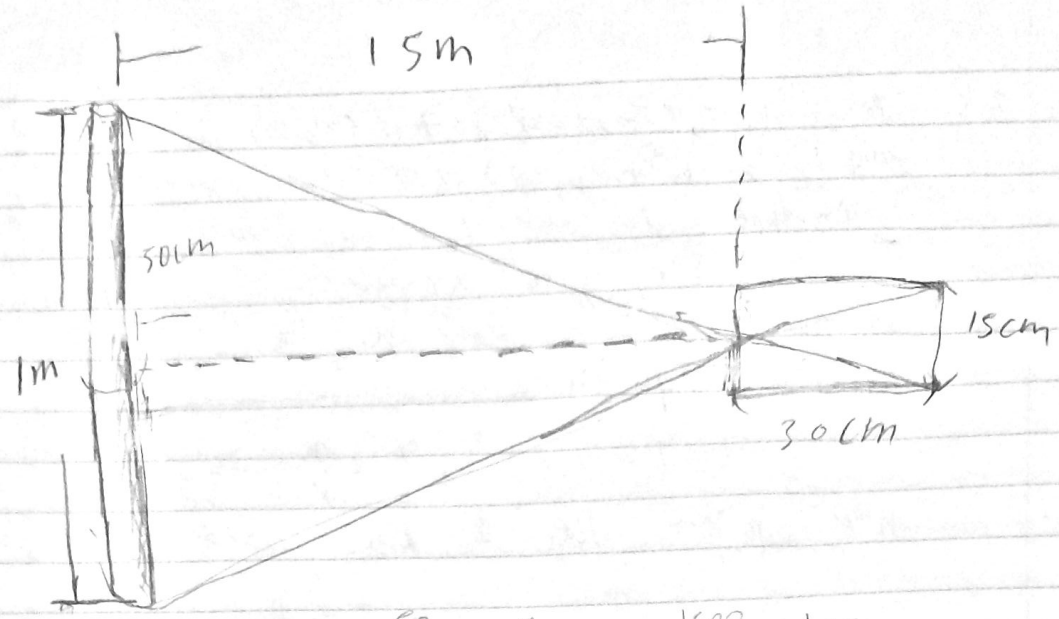
Then it shown



2.1)

$\frac{50}{1500} = \frac{a}{30}$ $a = \frac{1500}{1500} = 1\,cm$

$a \times 2 = 2\,cm$

2.2) object is 2 cm tall

2.3) $\frac{50}{d} = \frac{7.5}{30}$ $50 \cdot 30 / 7.5 = \frac{1500}{7.5} = 200\,cm$

Object at 200 cm or 2 m occupy entire screen.

2.4) suppose using light unit lux / mm^2

$a = \pi r^2$

$lux \cdot 10ms \cdot (1mm)^2 \pi = lux \cdot t_{ms} \cdot (10mm)^2 \pi$

$10 \cdot 1 = t \cdot 10^2 \Rightarrow$

$10^{-1} ms = t$

$t = 0.1\,ms$

$= 0.1$ millisecond

2.5 let $l_1 = \langle x_0, y_0, z_0 \rangle + t\langle a, b, c \rangle$, $c \neq 0$
and $l_2 = \langle x_1, y_1, z_1 \rangle + t\langle a, b, c \rangle$, $c \neq 0$

So that l_1 and l_2 are parallel lines in 3D space. while not parallel to the xy-plane.

let the projection plane be the xy-plane. with projection equation

$$(x, y, z) \to \left(-f\frac{x}{z}, -f\frac{y}{z}\right)$$

now project the 2 lines onto the xy-plane

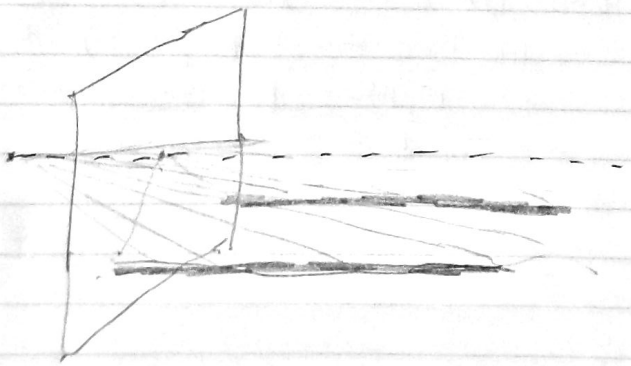
$$l_1 \Rightarrow (x, y, z) \to \left(-f\frac{x_0 + ta}{z_0 + tc}, -f\frac{y_0 + tb}{z_0 + tc}\right)$$

$$l_2 \Rightarrow (x, y, z) \to \left(-f\frac{x_1 + ta}{z_1 + tc}, -f\frac{y_1 + tb}{z_1 + tc}\right)$$

$$\lim_{t\to\infty} \left(-f\frac{x_0 + ta}{z_0 + tc}, -f\frac{y_0 + tb}{z_0 + tc}\right) = \lim_{t\to\infty} \left(-f\frac{x_1 + ta}{z_1 + tc}, -f\frac{y_1 + tb}{z_1 + tc}\right)$$

$$\left(-f\frac{a}{c}, -f\frac{b}{c}\right) = \left(-f\frac{a}{c}, -f\frac{b}{c}\right)$$

we just showed that the projection of l_1 and l_2 converge to the same vanishing point.



Tien Li Shen

February 22, 2021

Mini-project 1

Question 1

```
import numpy as np
import math

# Name: Tien Li Shen
# Homework 1 Q1

def pt1n2(m, n):
    pt1 = np.zeros((m,n))
    pt2 = np.random.rand(m,n)
    print(pt1)
    print(pt2)
    return (pt1, pt2)

def pt3(v): # compute norm
    nor = sum(sum(v**2))**(1/2) # element-wise square -> sum -> sum -> sqrt
    print(nor)
    return nor

def pt4(u ,v):
    dot = np.dot(u[0], v[0])
    angle = np.arccos(dot/(pt3(u)*pt3(v)))
    print(angle)
    euc_dist = (sum(sum((u-v)**2)))**(1/2)
    print(euc_dist)

def pt5(a):
    mn = a.shape[0] * a.shape[1]
    b = np.reshape(a, (mn, 1))
    print(b)


if __name__ == '__main__':
    m = 5
    n = 4
    v = np.array([[1, 2, 3, 4]])
    u = np.array([[4, 3, 2, 1]])

    print("pt 1 and 2")
    pt1n2(m, n)
    print("pt 3")
    pt3(v)
    print("pt 4")
    pt4(u,v)
    a = np.array([[1, 2, 3, 4], [4, 3, 2, 1]])
    print("pt 5")
    pt5(a)
```

Question 3

```
import numpy as np
import math

# Name: Tien Li Shen
# Homework 1 Q3

def pt1(F, S):
    # R = [[Rr1, Rr2, Rr3],
    #      [Rg1, Rg2, Rg3],
    #      [Rb1, Rb2, Rb3]]
    size = F.shape[0]
    R = np.zeros((size,size))
    for i in range(size):
        for j in range(size):
            R[i][j] = np.dot(F[j], S[i])
    return R

def pt2(R):
    C_turquoise = np.array([0.2509, 0.8784, 0.8156])
    C_goldenrod = np.array([0.8549, 0.6470, 0.1254])

    # turquoise
    R_inv = np.linalg.inv(R)
    b_turquoise = np.zeros(C_turquoise.shape[0])
    for i in range(C_turquoise.shape[0]):
        b_turquoise[i] = np.dot(R_inv[i], C_turquoise)

    # goldenrod
    b_goldenrod = np.zeros(C_goldenrod.shape[0])
    for i in range(C_goldenrod.shape[0]):
        b_goldenrod[i] = np.dot(R_inv[i], C_goldenrod)

    return [b_turquoise, b_goldenrod]


if __name__ == '__main__':

    # F => 3 flash lights emssion spectral distribution
    F = np.array([[0.00, 0.00, 0.00, 0.00, 0.01, 0.02, 0.07, 0.29, 0.35, 0.12],
    [0.00, 0.01, 0.02, 0.06, 0.20, 0.31, 0.20, 0.16, 0.04, 0.00],
    [0.03, 0.15, 0.25, 0.27, 0.12, 0.02, 0.01, 0.01, 0.00, 0.00]])

    # S => Human eye RGB absorption distribution
    S = np.array([[0.16, 0.26, 0.28, 0.15, 0.10, 0.03, 0.02, 0.00, 0.00, 0.00],
    [0.00, 0.03, 0.06, 0.20, 0.31, 0.21, 0.15, 0.03, 0.01, 0.00],
    [0.00, 0.00, 0.00, 0.00, 0.01, 0.04, 0.08, 0.23, 0.35, 0.29]])
```

```
    R = pt1(F, S)
    print("PT1\nR = ", R)
    b_turquoise, b_goldenrod = pt2(R)
    print("PT2\nb_turquoise = {}\nb_goldenrod = {}".format(b_turquoise,
b_goldenrod))
```

Question 4

```
import numpy as np
from matplotlib import pyplot as plt
from PIL import Image

# Name: Tien Li Shen
# Homework 1 Q4

def grayworld(im): #[L, C] = grayworld(I)
    I = np.array(im)
    dim = I.shape
    I_avg = np.zeros(3)

    #calculate the average I_avg values
    for i in range(dim[0]):
        for j in range(dim[1]):
            for k in range(len(I_avg)):
                I_avg[k] += I[i][j][k]
    I_avg = I_avg/dim[0]/dim[1]

    L = I_avg/np.array([128, 128, 128])

    #get the 128/r, 128/g, 128/b ratios
    rgb_ratio = np.array([128, 128, 128]) / I_avg

    #get color image C
    C = np.zeros(dim)
    for i in range(dim[0]):
        for j in range(dim[1]):
                C[i][j] = I[i][j] * rgb_ratio

    # scale pixels to max of 255
    C = C/np.amax(C)*255

    C = Image.fromarray(np.uint8(C))
    return [L, C]

if __name__ == '__main__':
    im = Image.open(r"data/wb_sardmen-incorrect.jpg")
```

```
L, C = grayworld(im)
print("L = ", L)

# plot the image
plt.subplot(1,2,1)
plt.imshow(im)
plt.title("Original")
plt.subplot(1,2,2)
plt.imshow(C)
plt.title("Color Corrected")
plt.show()
```