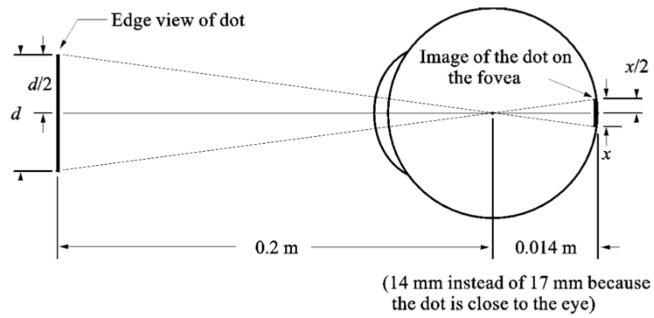
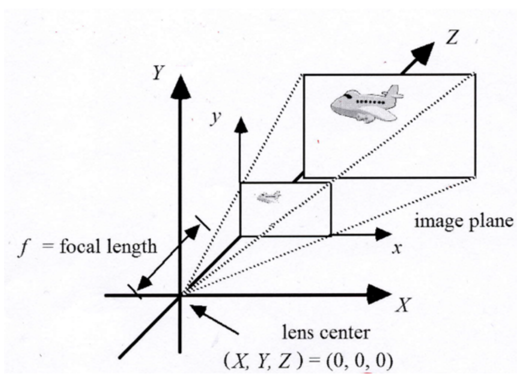
ELET 6397 Spring 2019
Homework 01
Due Thursday Feb 14 at 11:59 pm

1. Use the image formation geometry discussed in class, estimate the diameter of the smallest printed dot that the eye can discern if the page on which the dot is printed is 0.2m away from the eyes. Assume for simplicity that the visual system ceases to detect the dot when the image of the dot on the fovea becomes smaller than the diameter of one receptor (cone) in that area of the retina. Assume further that the fovea can he modeled as a square array of dimension 1.5 mm X 1.5 mm, and it has 336,400 cones and spaces between the cones, which are both distributed uniformly throughout this array.



2. At the air traffic control center at Houston airport, an United Airline plane is tracked by radar and is also imaged by an optical camera with a focal length f = 1 cm interfaced with a computer. For this camera shown below is the imaging geometry with 3-D world coordinates (X Y, Z) using the upright projection model as defined in class.



Answer the following questions, assuming that the image of the jet is a single point

(a) At time t = 0 seconds, the image of the airplane is optically detected at point

$$p = (a_0, b_0) = (0.4, 0.5)\ mm$$

What locus of 3D world points (A_0, B_0, C_0) must the airplane lie on if it has a range $C_0 \geq 1\ km$?

(b) The radar determines that in fact the range of the airplane is $C_0 = 5\ km\ at\ time\ t = 0.$
At time t = 5 seconds, the radar reports that the airplane is still at range $C_5 = 5\ km$, and the optical camera records the position of the image of the jet to be $(a_5, b_5) = (0.8, 1.0)\ mm.$ How far did the airplane travel in real 3D space during the time interval [0, 5] seconds?

(c) What was the average real speed of the airplane during the interval [0, 5]?

Basic Image Operations

3. You are given a JPEG image ***Zootopia.jpg***, and will perform some basic image operations on it (Helpful functions: ***imread, imshow, figure, title, subplot, double, uint8, size***).

Zootopia.jpg

(a) Read the ***Zootopia.jpg*** image eparate its three color-components – i.e., R (red), G (green), and B (blue). Show the original image and the three color-components in a 2x2 grid with labels.

(b) Generate two images from the original image: one with the G and B components exchanged, and the other with the R and G components exchanged. Show them side-by-side.

(c) YUV is an older analog color space. It defines brightness in terms of wavelength=sensitivity. Convert= the original image to YUV format, i.e. generate three images of Y, U and V components separately by using the converting equation described in the lecture slide . Show them side-by-side.

(d) The Y component image is actually the luminance image of the original one. Generate another Y component image by linearly inverting the pixel value (i.e., 0 to 255 and 255 to 0)=of the original Y component image. Show both the original and inverted Y component images side-by-side with labels (Hint: be careful about the data type, e.g., uint8 and double).

4. Bit-Planes

You are given two images, ***Transformers.jpg*** and ***Transformers-msg.jpg***, and will find a secret image hidden in the bit-planes (Helpful functions: ***bitget***, ***bitset***).

Transformers.jpg

(a) Read both images and show them side-by-side. Can you distinguish the difference between them and identify the hidden message in ***Transformers-msg.jpg***?

(b) Extract the 8 bit-planes of ***Transformers-msg.jpg*** using ***bitget***. Display the extracted bit-planes in a 4x2 grid and properly label each bit-plane. Here, we assume bit-plane 1 is the least significant bit (LSB) and bit-plane 8 is the most significant bit (MSB).

(c) Which planes are visually significant? What is the message hidden in ***Transformers-msg.jpg***?This problem is related to the concept of digital image watermarking. Digital image watermarking is a technique in which the owner of a digital image embeds a perceptually transparent "ownership signature" within an image.

(d) Now let's figure out how many messages can be stored in the original ***Transformers.jpg*** before its quality becomes unacceptable. Replace bit-plane 1 of the original image with the hidden image using ***bitset***. Repeat this step for bit-planes 2 through 8. Show the new images in a 4x2 grid with labels. Which images are perceptually distorted?

5. Create three 3 x 3 ”images” having only 0 and 1 as elements. These images will be the red, green, and blue component images of a 3 x 3 color image that you will display. The color image should look like this:

white	red	blue
green	yellow	magenta
cyan	white	black

Figure 1: Desired 3 × 3 image.

For example, the red image could be created by the statement

```
>> r=[0,1,0;0,1,1;0,1,1];
```

(a) After you have created the `g` and `b` images, use the Matlab statements

```
>> rgb_image=cat(3,r,g,b);
>> imshow(rgb_image);
```

to display the image. Look at the image carefully to confirm that you have placed the saturated primary colors RGB and CMY in the correct places. Print the values of the three images by using the Matlab command

```
>> [r,g,b]
```

The resulting image plot is rather small (3 x 3 pixels). Use the magnification feature of the `imshow` command to scale the color image to a larger size for viewing.

Give the Matlab *statements used to generate the green and blue images.*
Also include the scaled image plot to show that you have the correct component arrays

(b) Scale each of the images by a constant that is less than one and display it; e.g., scaling by 0.8 is implemented as:

```
>> c=0.8;
>> rgb_scaled=cat(3,r*c,g*c,b*c);
>> imshow(rgb_scaled)
```

Use values of `c = 0.8, 0.6, 0.4, 0.2` to observe the effect.
What happens to the colors as `c` *is varied over the range of values?*

6. Load and display the color image `zootopia.jpg`.

(a) Convert the class `uint8` color image to a gray scale image (`rgb2gray command`), and display the full intensity range gray-scale image using the `imshow` command.

(b) Now using `imshow` change the intensity range [0 255] to a lower range [0 `N`] for N = 200, 150, 125 and 100. *What value of* `N` *begins to distort the image? Why?*

(c) Test the `imcrop` command out on a display of the grayscale image from (a) above. Crop out the head of the sloth and display it on the screen using `imshow`. *Describe how you used the* `imcrop` *to crop sloth head. Give position (X,Y) for the begin and end of the crop?*

(d) Plot the intensity profile at row 180 (see command `plot`).

(e) Another way to display an RGB image is to convert it to an indexed image with its associated color map, and then plot it using `imshow`. The conversion is done using the Matlab function `rgb2ind`. The method of computing the color map depends on the number and type of arguments given. Use `help rgb2ind` to see the possibilities. For example `rgb2ind(image);` generates a map containing entries for every pixel in the image, while `rgb2ind(image,N);` generates a colormap of `N` entries by a process that "best approximates the original RGB image." Try different values for `N` and observe the effects. Try plotting the color maps using `rgbplot(map)`. *How many colors give a visually pleasing image? As N decreases, what effects of color quantization do you see?*

(f) Convert the color image `'zootopia.jpg'` from RGB to YIQ format using the formulas provided in class slides and also the Matlab command `rgb2ntsc`. Are the two images the same.

(g) Using the mutliplot capability of subplot, plot the R, G, B and RGB composite components in a 2 x 2 image plot format. Plot the Y, I, Q components of the image (along with the composite RGB image) in another 2 x 2 image plot.

(h) Recalling that R, G, and B are intensity images such that $0 \leq R \leq 1$, $0 \leq G \leq 1$, and $0 \leq B \leq 1$, determine the ranges of the YIQ variables. These ranges are important, since only if Y, I, and Q satisfy them can we guarantee that the inverse transformation will give values of R, G, and B in the correct ranges. *Give the ranges determined for Y, I, and Q.*

(i) Convert the YIQ image back to RGB format using the Matlab command `ntsc2rgb(yiq_image)` and plot the reconverted R, G, B components, along with the composite RGB image on another 2 x 2 image plot. How close to the original RGB components does the double transformation come?

(j) Compute the differences between corresponding images and use the `max` function to determine the maximum error in the conversion. *What was the size of the errors?*