CV Image Basic Operations
Allen Lau

I. Writing Assignments

1. How does an image change (e.g., objects' sizes in the image, field of view, etc.) when you change the zoom factors of your pinhole camera (i.e., the focal length of a pinhole camera is changed)?

   The focal length, defined as the distance between the pinhole lens and image plane, of the pinhole camera affects both the orientation and the size of the image. For example, if the focal length distance is positive, then the image of the object will appear inverted. As the image plane moves closer to Z=0, the image becomes smaller. As the image plane moves towards larger magnitude negative Z values, the image becomes larger. When the image plane is located at a positive Z value, the object in the image will match the orientation of the real-world object, getting larger as the Z value increases in magnitude. This is described by a negative focal length distance.

2. Give an intuitive explanation why a pinhole camera has an infinite depth of field, i.e., the images of objects are always sharp regardless their distances from the camera.

   A pinhole camera has an infinite depth of field because there is no bending of light towards a focal point, like with thin lens model cameras. This means light emitted from the objects and entering the pin hole will travel in straight lines through the pinhole and will intersect with the image plane. Additionally, multiple light rays from the same object point do not enter the pin hole (only one ray for that point will go through the hole), so there is not a distance where the light from the same point intersects after entering the camera. This results in an in focused but inverted image, assuming the focal length is positive.

3. In the thin lens model, 1/o + 1/i = 1/f, there are three variables, the focal length f, the object distance o and the image distance i (please refer to the Slides of the Image Formation lecture). If we define Z = o-f, and z = i-f, please write a few words to describe the physical meanings of Z and z, and then prove that Z*z = f*f given 1/o + 1/i = 1/f.

   z = i – f represents the distance between the image plane to the focal point of the thin lens. Z = o – f represents the distance between the object and the focal point, since in a thin lens model, there are two focal points. For this equation, f represents the focal point closest to the object.

   Proof that Z * z = f * f given 1/o + 1/i = 1/f

$$\frac{1}{o} + \frac{1}{i} = \frac{1}{f}, Z = o - f, z = i - f$$

$$\frac{1}{Z+f} + \frac{1}{z+f} = \frac{1}{f}$$

$$\left(\frac{z+f}{z+f} * \frac{1}{Z+f}\right) + \left(\frac{Z+f}{Z+f} * \frac{1}{z+f}\right) = \frac{1}{f}$$

$$\frac{z+f}{(Z+f)(z+f)} + \frac{Z+f}{(Z+f)(z+f)} = \frac{1}{f}$$

$$\frac{z+f+Z+f}{(Z+f)(z+f)} = \frac{1}{f}$$

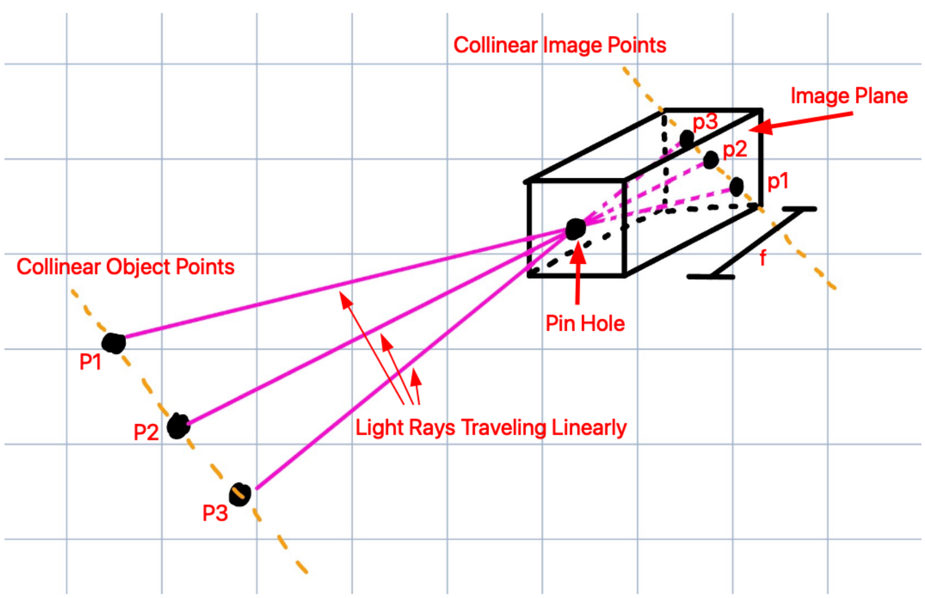
$$f * (Z+f)(z+f) * \left(\frac{z+f+Z+f}{(Z+f)(z+f)}\right) = \left(\frac{1}{f}\right) * f * (Z+f)(z+f)$$

$$2f^2 + zf + Zf = Zz + Zf + zf + f^2$$

$$f^2 = Z\,z$$

$$Z * z = f * f$$

4. Prove that, in the pinhole camera model, three collinear points in the world (i.e., they lie on a line in 3D space) are imaged into three collinear points on the image plane. You may either use geometric reasoning (with line drawings) or algebra deduction (using equations).



Three collinear points in the world (as denoted by P1, P2, and P3 above) will be imaged into three collinear points on the image plane (as denoted by p1, p2, and p3) because there will be a single linearly traveling light ray from each of the world object points that enters the pin hole camera. Since there is no bending or diffraction with a pin hole camera, the light continues to travel in that straight path until it intersects the image plane. As seen in the drawing above, this geometric path results in the image points (p1, p2, and p3) being collinear but the image is inverted when compared to the object points.

II. Programming Assignments

Image formation. In this small project, you are going to read, manipulate and write image data. The purpose of the project is to make you familiar with the basic digital image formations. Your program should do the following things:

1. Read in a color image C1(x,y) = (R(x,y), G(x,y), B(x,y)) in Windows BMP format, and display it.

Figure 1: ID Picture

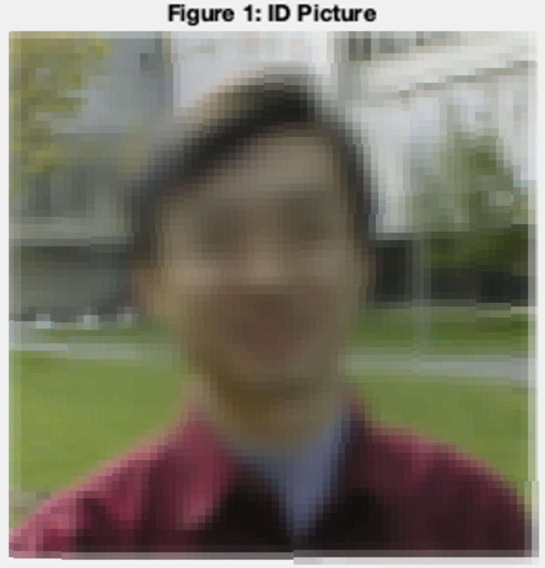


Figure 1 displays a Windows BMP format image that is 250 x 250 x 3, where the first two numbers indicate the number of rows and columns and the 3 represents the red, green, blue color channels. From visual inspection, we can see that the scene is dominated by greens from the grass, reds from the shirt, and white-like colors from the background.

2. Display the images of the three-color components, R(x,y), G(x,y) and B(x,y), separately. You should display three black-white-like images.

Figure 2: ID Picture and 3 Color Components

Figure 2 displays the original ID image, as seen in Figure 1, in the upper-left corner. The upper-right corner displays the red component image (R(x,y)), the bottom-left displays the green component (G(x,y)), and the bottom-right displays the blue component (B(x,y)).

For each component image, the pixel intensity values for corresponding color component in the original image are copied into all three channels. This will create a black-white-like image, which gives information about where there is more of a certain color. For instance, in the green component image, the grass looks lighter (closer to intensity value 255 in all three channels of the G(x,y) image at the x,y locations where there is grass) since grass is green. Additionally, areas of the red shirt closer to pure red (areas not in shadows) in the original image look close to white in the red component image. Lastly, since white is created when R=255, G=255, and B=255 and black is created when R=0, G=0, and B=0, areas of the original image that are white or black still appear white or black in the color component images.

3. Generate an intensity image I(x,y) and display it. You should use the equation I = 0.299R + 0.587G + 0.114B (the NTSC standard for luminance) and tell us what are the differences between the intensity image thus generated from the one generated using a simple average of the R, G and B components. Please use an algorithm to show the differences instead by just observing the images by your eyes.

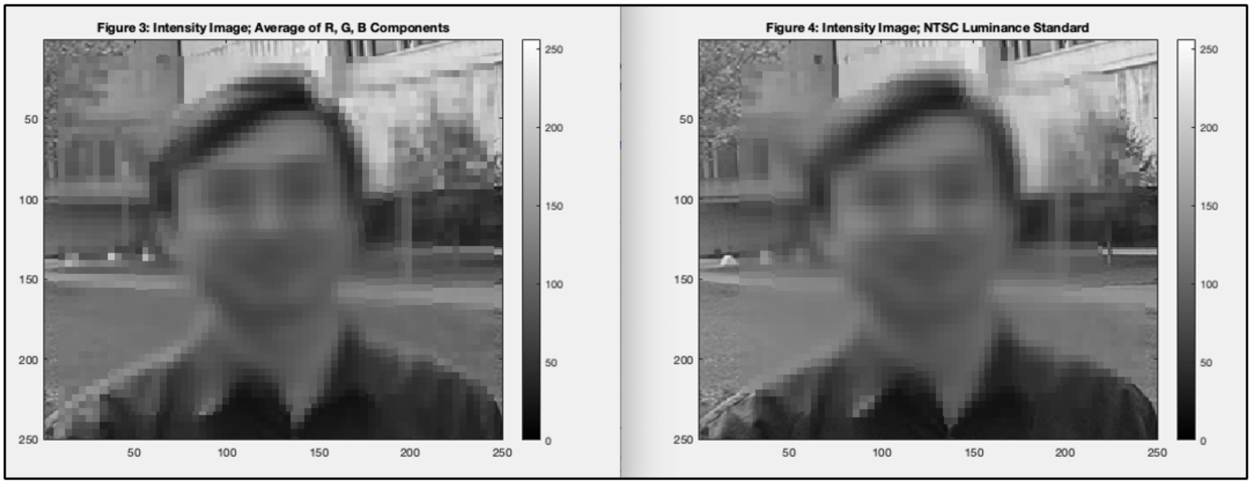
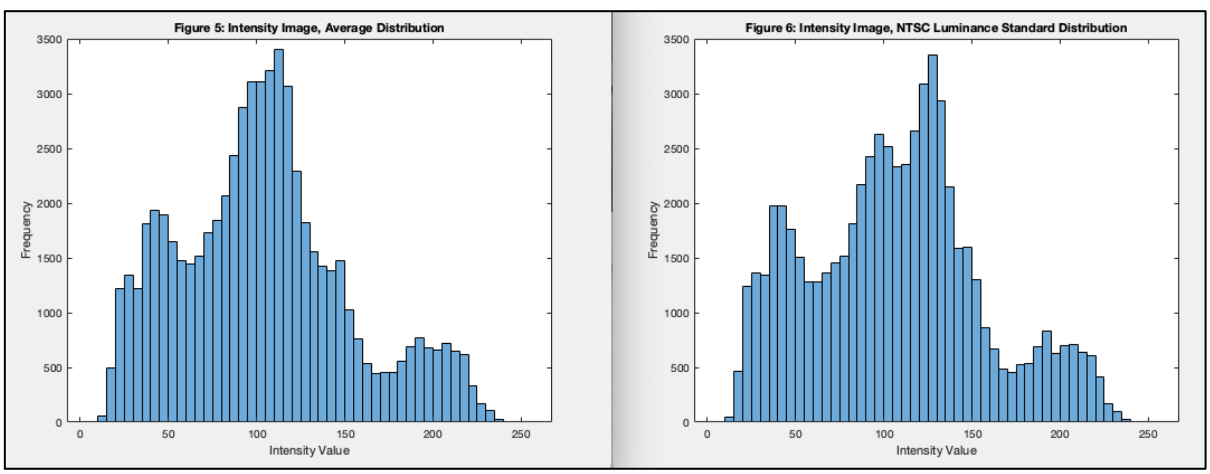


Figure 3 displays the 250 x 250 x 1 intensity image computed via taking the average of the three color components. Figure 4 displays the 250 x 250 x 1 intensity image computed via the NTSC Luminance Standard.



Using the above histograms, we can quantitatively see that the two intensity images are similar due to the distributions following relatively the same curve. However, intensity values near the middle (~75 to ~125) of the image derived from the simple average have shifted to the right and has split from one prominent mode to two modes in the NTSC derived image. This is due to how the NTSC equation weights the different color components differently (with green components being weighted the highest and blue components being weighed the least), in contrast to the equal weighting of the simple average.

4. The original intensity image should have 256 gray levels. Please uniformly quantize this image into K levels (with K=4, 16, 32, 64). As an example, when K=2, pixels whose values are below 128 are turned to 0, otherwise to 255. Display the four quantized images with four different K levels and tell us how the images still look like or different from the original ones, and where you cannot see any differences.

Figure 7 displays the uniformly quantized intensity images into K levels, where K = 4, 16, 32, 64. At K=4, it is easily noticed that the image is segmented into distinct groups of pixels and is quite different from the original image. At K=16, you are still able to notice that the intensity values are being quantized, since there are rough transitions of grayscale intensity from one pixel to a neighboring pixel at certain parts of the image. This rough transition is because there are less values of intensity that are being displayed to show smooth transitions of grays from the original image. At K=32 and K=64, it now becomes difficult to discern the difference between the two. As a result, for this example intensity image at a resolution of 250x250, K=32 is the quantization level where we are unable to see any differences between the original and the quantized image.

5. Quantize the original three-band color image C1(x,y) into K level color images CK(x,y)= (R'(x,y), G'(x,y), B'(x,y)) (with uniform intervals) , and display them. You may choose K=2 and 4 (for each band). Do they have any advantages in viewing and/or in computer processing (e.g., transmission or segmentation)?

Figure 8 displays the uniformly quantized color image at K=2, 4. It can be observed that there are advantages to viewing images via quantization. For instance, feature extraction and segmentation are advantages since quantization will group similar pixels together. In this example, at K=2, the wall of the building is composed of a majority color instead of many different shades of colors like in the original image or even the two levels in the K=4 image above. At K=2, it is likely easier to segment that part of the image. Another advantage is compression and memory. At low K values, the number of bits needed to represent the pixel intensity values decreases. For instance, at K=2, 1 bit is needed to represent the 2 levels and at K=4, 2 bits are needed. In contrast, at K=256, 8 bits are required. This results in a reduced amount of memory needed to be allocated for these images; therefore, less disk space is required to store the images and transmission of the image is faster.

6. Quantize the original three-band color image C1(x,y) into a color image CL(x,y)= (R'(x,y), G'(x,y), B'(x,y)) (with a logarithmic function) , and display it. You may choose a function I' =C ln (I+1) (for each band), where I is the original value (0~255) , I' is the quantized value, and C is a constant to scale I' into (0~255), and ln is the natural logarithmic function. Please describe how you find the best C value so for an input in the range of 0-255, the output range is still 0 – 255. Note that when I = 0, I' = 0 too.

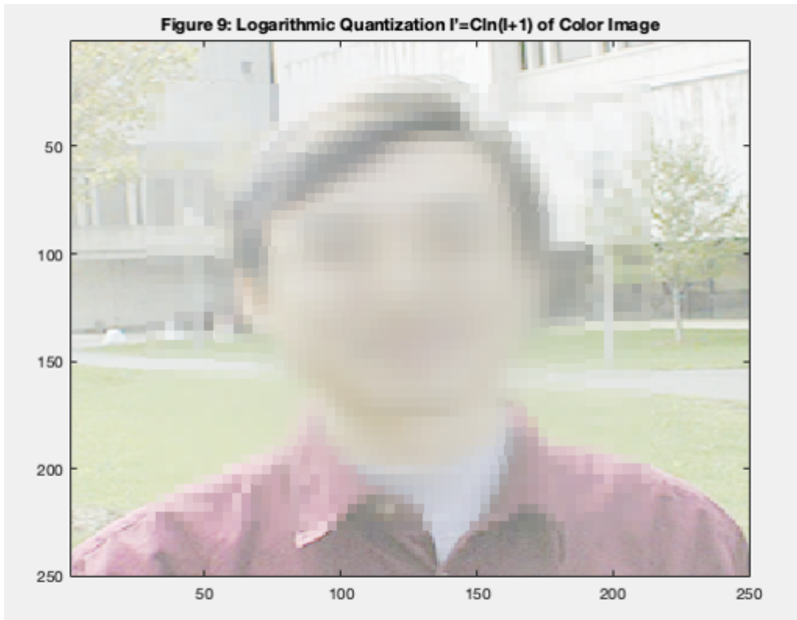


Figure 9 above depicts the quantized color image using the logarithmic function I' = Cln(I+1). It is observed that the image is much lighter than the original. This is due to the quantization function being logarithmic where as I increases from 0 to 255, the rate by which I' increases, decreases at a faster rate. As a result, there is a larger range of intensity values, I, that is mapped to a smaller range of intensity values, I', that is closer to 255.

To find the best value of C for mapping an input in the range of 0 – 255 to an output range of 0 – 255 for this equation, we set I' = I = 255 and solve for C. Note, for the mapping of I = 0, the equation simplifies to I'=Cln(1) which is always 0.

Finding C to map input range of 0 – 255 to output range of 0 – 255:

$$I' = Cln(I+1)$$
$$C = \frac{I'}{\ln(I+1)}\ , set\ I' = I = 255$$
$$C = \frac{255}{\ln(255+1)} = \frac{255}{\ln(256)} \approx 45.9859$$