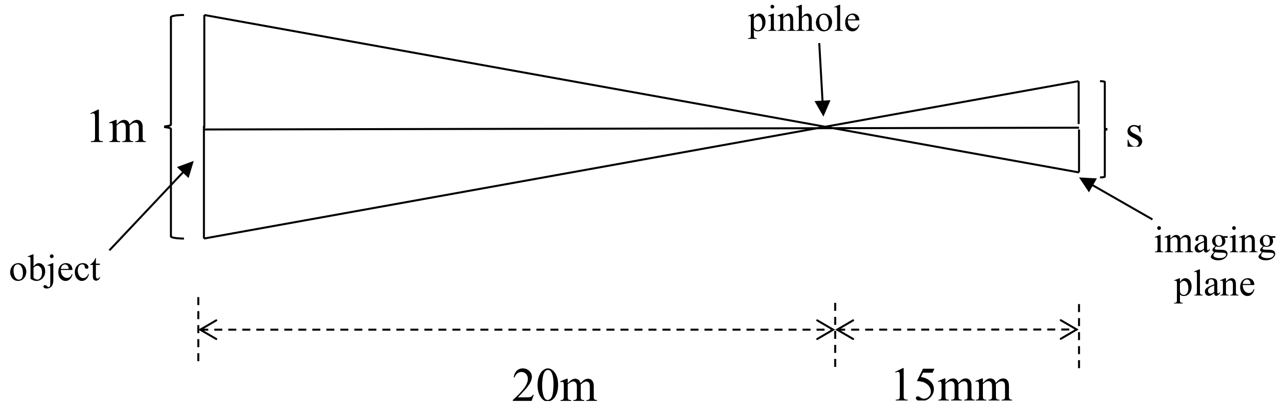
Homework #1
Due Mon. Oct. 3 11:59pm through CatCourses
(Upload a PDF of a legible scan of your written solution or a PDF of it typeset in Word/Latex/etc.)

1. In this problem, you will determine the size the image of an object forms on the imaging plane, assuming a pinhole camera.

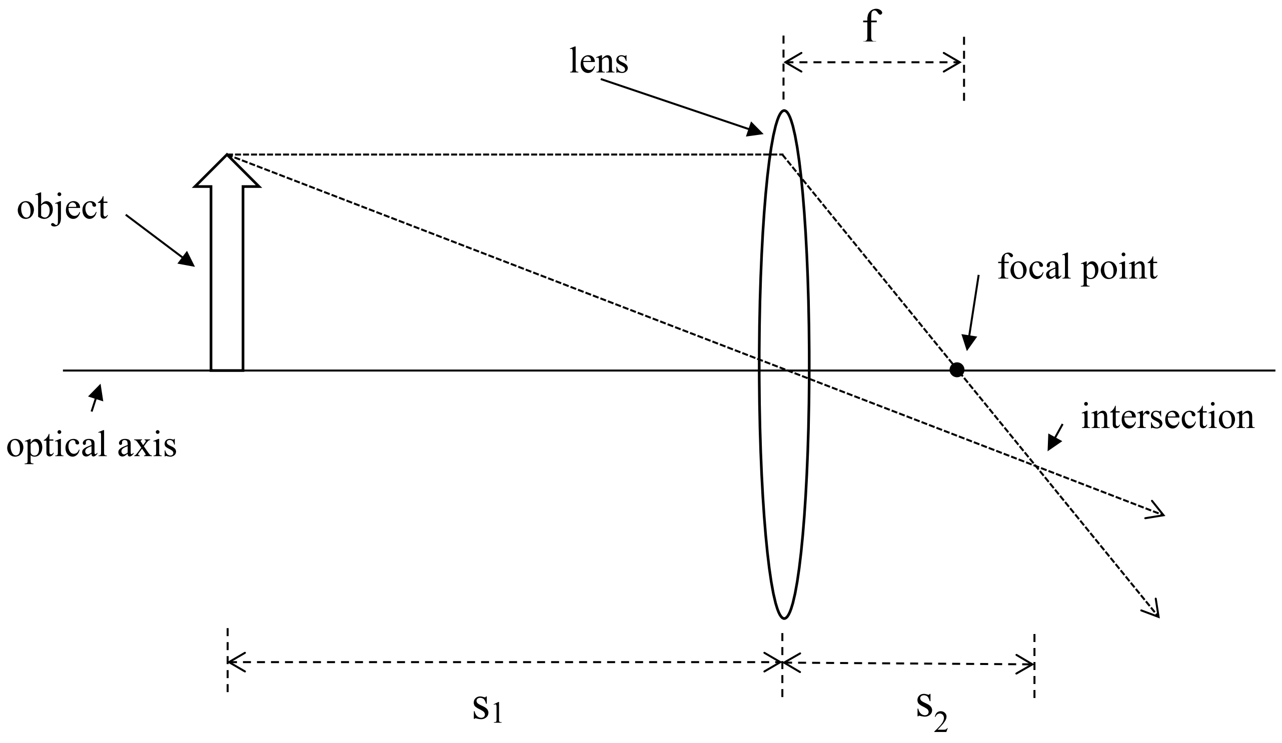
   See the figure below in which an object of size 1m is at a distance of 20m from a pinhole camera. The imaging plane is 15mm from the pinhole.



   For this particular configuration, determine s.

2. In this problem, you will determine the correct distance the imaging plane should be from the lens for an object at a certain distance to be in focus, assuming a thin lens model.

   See the figure below which shows the geometry of a thin lens model. Light travelling parallel to the optical axis is refracted by the lens through the focal point which is at a distance f from the lens. Light which hits the center of the lens is not refracted but continues traveling straight. If the imaging plane is placed where these two intersect then the object will be in focus.



a) For the thin lens model, derive the equation which relates the following three quantities:

   - f: the focal length of the lens.
   - s_1: the distance that the object is from the lens.
   - s_2: the distance between the lens and the intersection of the rays of light from the object.

b) Suppose that f=25cm and that the object is 10m away. At what distance, s_2, should the imaging plane be from the lens in order for the object to be in focus?

c) Suppose the object moves further away from the lens. Should the imaging plane be moved further or closer to the lens in order for the object to remain in focus?

3. In this problem, you will calculate the cost of transmitting various forms of multimedia across a mobile data channel.

   Assume that your mobile data plan costs you $4 for 2GB (where 1GB is 1,024MB, 1MB is 1,024 KB, and 1KB is 1,024 bytes). Assume, also, that you pay proportionally. That is, if your usage is not an increment of 2GB then you pay only for the fraction that you use (whether it is less or more than 2GB).

   a) Calculate how much it would cost for you to download all of Shakespeare's plays where the total word count of all his plays is 835,997. Assume the average word length is five letters and that you represent each letter using a byte (we are ignoring spaces, punctuation, and formatting).

   b) Calculate how much it would cost you to download a single image from a Canon EOS 5D Mark III DSLR camera without any compression.

      This camera produces 22.3 mega-pixel images where

      $$1\,mega - pixel = 1{,}024 \times 1{,}024\,pixels$$

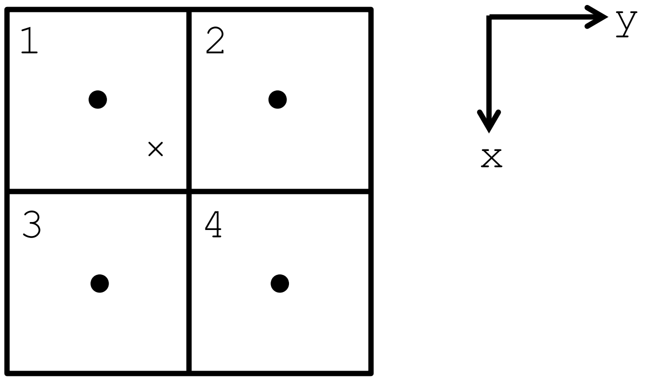
      and each pixel is represented using three bytes, one for each of the red, green, and blue color channels.

   c) Calculate how much it would cost to download a one-hour ultra high-definition 4k video without compression.

      A 4k video has 60 frames per second where each frame measures 3,840 by 2,150 pixels. Again, assume each pixel is represented using three bytes.

4. This problem investigates nearest neighbor and bilinear interpolation. For simplicity, we will focus on estimating the image intensity at a single location. Interpolation is used when transforming an image through resizing, rotating, etc., in which case, the image intensity will need to be estimated at a number of locations.

Consider the diagram below of four pixels



where the dots (●) represent the locations where we know the image intensity and the × represents the location where we would like to estimate the image intensity. By convention, the vertical axis is the x-axis and the horizontal axis is the y-axis.

Suppose the four pixels are at the following locations (indicated by (*x*,*y*)) and have the following intensity values (indicated by *p*)

$$\begin{aligned} (x_1, y_1) &= (4,10) \qquad p_1 = 20 \\ (x_2, y_2) &= (4,11) \qquad p_2 = 25 \\ (x_3, y_3) &= (5,10) \qquad p_3 = 200 \\ (x_4, y_4) &= (5,11) \qquad p_4 = 250 \end{aligned}$$

and that we would like to estimate the image intensity at a fifth location

$$(x_5, y_5) = (4.2,10.2)$$

That is, we want to estimate p_5.

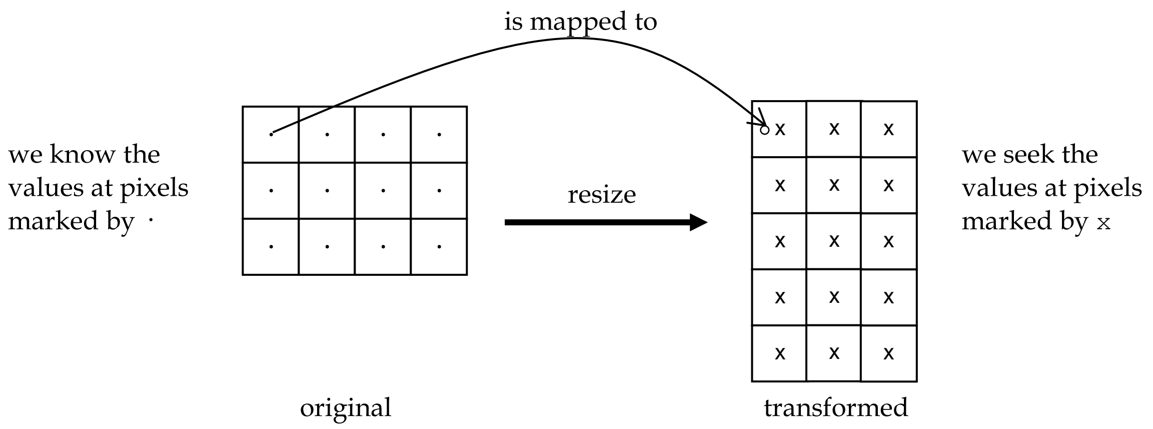
a) Provide an estimate for p_5 using nearest neighbor interpolation.

b) Provide an estimate for p_5 using bilinear interpolation. Round your value to the nearest integer. You can use either of the two methods discussed in lecture. (You might want to use both methods to check your answer.)

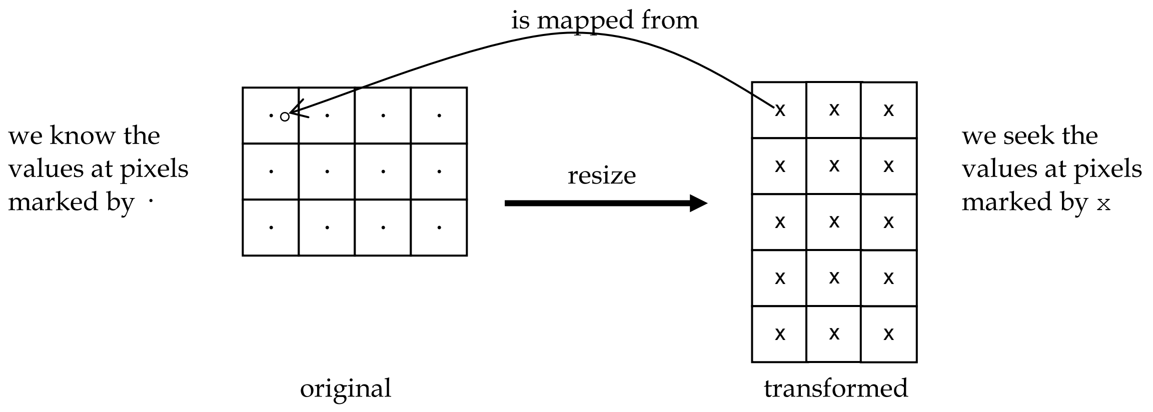
5. This problem will help with your lab project on image resizing.

   Geometric transformations of an image, such as resizing, produce images whose pixels typically don't align with pixels in the original image. This raises the question of what values to assign to the pixels in the transformed image. This can be accomplished using interpolation. Most interpolation approaches estimate the pixel values in the transformed image using the values of the "closest" pixels in the original image.

   Note that rethinking the transformation helps with solving this problem. All we really want are the pixel values in the transformed image. So, rather than computing where the pixels in the original image are mapped to, such as in the diagram below,


is mapped to
we know the values at pixels marked by ·
resize
we seek the values at pixels marked by x
original
transformed

   we instead compute where the pixels in the transformed image are mapped from, as in the diagram below,

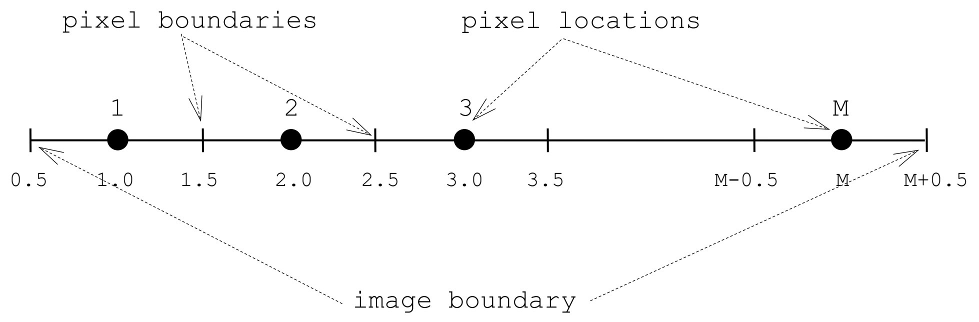

is mapped from
we know the values at pixels marked by ·
resize
we seek the values at pixels marked by x
original
transformed

   Thus, we actually perform the inverse mapping.

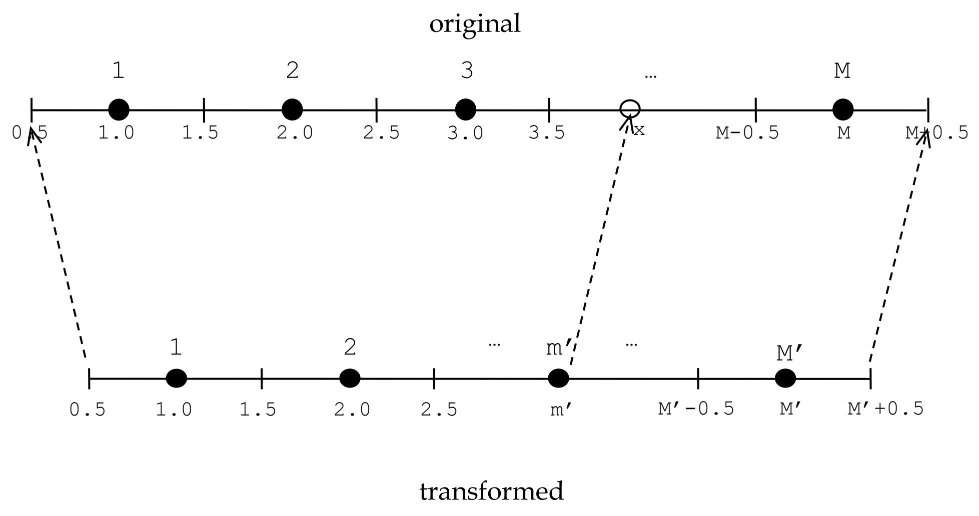
Once we have computed where the pixel is mapped from, we can use the values of surrounding pixels in the original image (whose values we know) to estimate the value of the pixel in the transformed image. Nearest neighbor interpolation uses the value of the closest pixel. Bilinear interpolation uses the values of the four closest pixels. In this problem, you will explore how to determine the *locations* of these pixels (not the values).

To simplify the problem, we will work in 1D. Instead of determining the closest pixels in an image, we will only consider the closest pixels on a line. The extension to 2D should be straightforward. We will also only consider resizing as the geometric transformation.

We will model a 1D M-pixel "image" as follows



Now, the inverse mapping of a pixel in a transformed image of size M′ to an original image of size M can be viewed as follows



a) Find the linear mapping from a location m′ in the transformed image to the location x in the original image. That is, derive a function that given m′, computes x. The transformed image measures M′ pixels and the original image measures M pixels. Note that M′ can be greater than or less than M.

Your mapping should have the following form

$$x = t(m') = \frac{A}{B}\left(m' - C\right) + D$$

where the constants A, B, C, D are determined from the following constraints:

- The left boundary of the transformed image should map to the left boundary of the original image, i.e., 0.5 should be transformed to 0.5.
- The right boundary of the transformed image should map to the right boundary of the original image, i.e., M′+0.5 should be transformed to M+0.5
- Locations in between should be mapped linearly (proportionally).

Note that A, B, C, D can be in terms of M and M′.

(One way to check if you have the correct equation is to see whether it satisfies the first two constraints above.)

Note that the linear equation above really only has two constants and could be written as

$$x = t(m') = Am' + B\,.$$

I provided the four-constant version above to help you think about how to incorporate the constraints (that form helps me, at least). You can use either form in your answer.

Note that x typically won't be integer valued (that is, it won't fall on a pixel location in the original image) even if m′ is integer valued.

b) Derive the logic and computation to determine the closest pixel m to a location x in the original image. That is, given x, determine m where m $\in$[1,…,M]. You can assume

$$0.5 \le x \le M + 0.5\,.$$

c) Derive the logic and computation to determine the two closest pixels m1 and m2 to a location x in the original image. Some special cases you might need to consider:

- x is integer valued
- x is less than 1.0
- x is greater than M
- x is equal to 1.0
- x is equal to M

d) Now, think about the 2D case. Both dimensions will need to be mapped from the transformed image to the original image. Nearest neighbor interpolation still only requires the closest pixel. Bilinear interpolation now requires the four closest pixels.

Suppose mapping a pixel in the transformed image along the m dimension results in the locations m1 and m2 being the m-coordinates of the closest points, and that mapping a pixel in the transformed image along the n dimension results in the locations n1 and n2 being the n-coordinates of the closest points. List the coordinates of the four closest points in 2D. (This should be fairly straightforward but I want to get you thinking about the problem in 2D for your project).