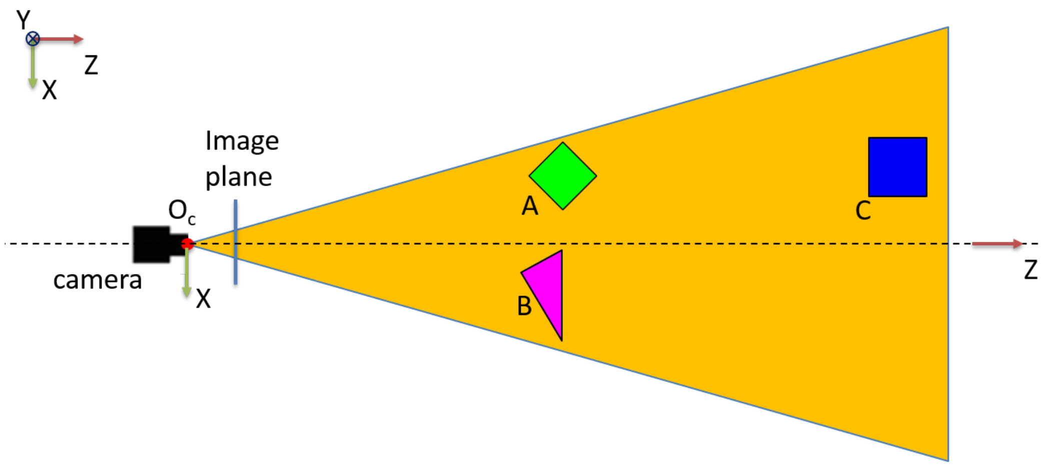
Dolly Zoom: Compute focal length

1) Introduction

In this lab, we will implement Dolly Zoom effect used by film-makers to create a sensation of vertigo, a ``falling-away-from-oneself feeling''. This lab is fairly simple and is meant to introduce you to the concepts of projection and focal length. It keeps the size of an object of interests constant in the image, while making the foreground and background objects appear larger or smaller by adjusting focal length and moving the camera. You will simulate the Dolly Zoom effect with a synthetic scene as shown in next Figure, which illustrates two cubes and one pyramid seen from the top view.



2) Dolly Zoom Effect

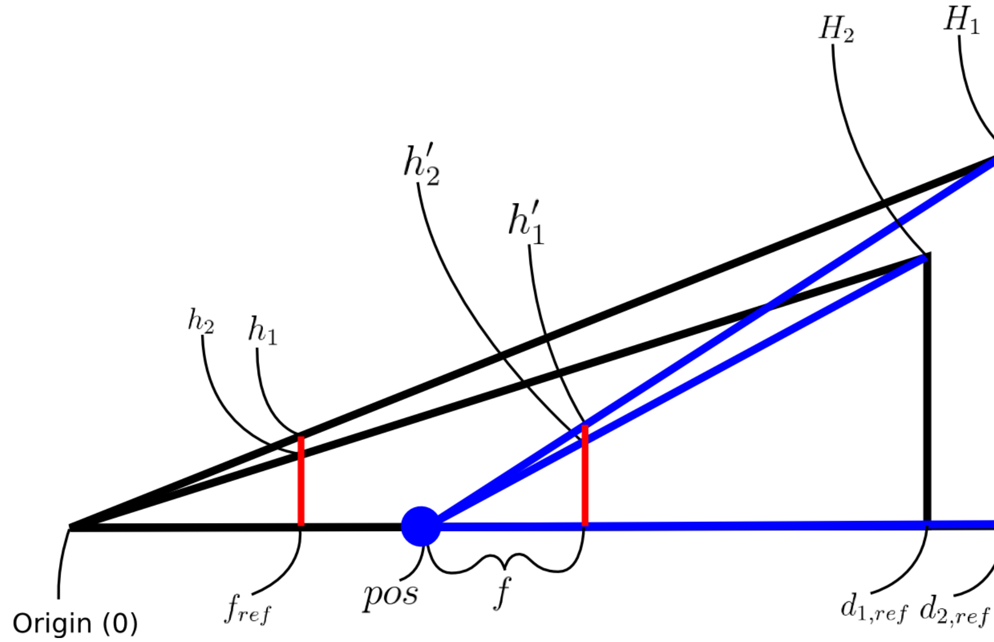
A point in 3D is projected onto the image plane through the pinhole (center of projection, COP):

$$u = f\frac{X}{Z}, \qquad v = f\frac{Y}{Z}$$

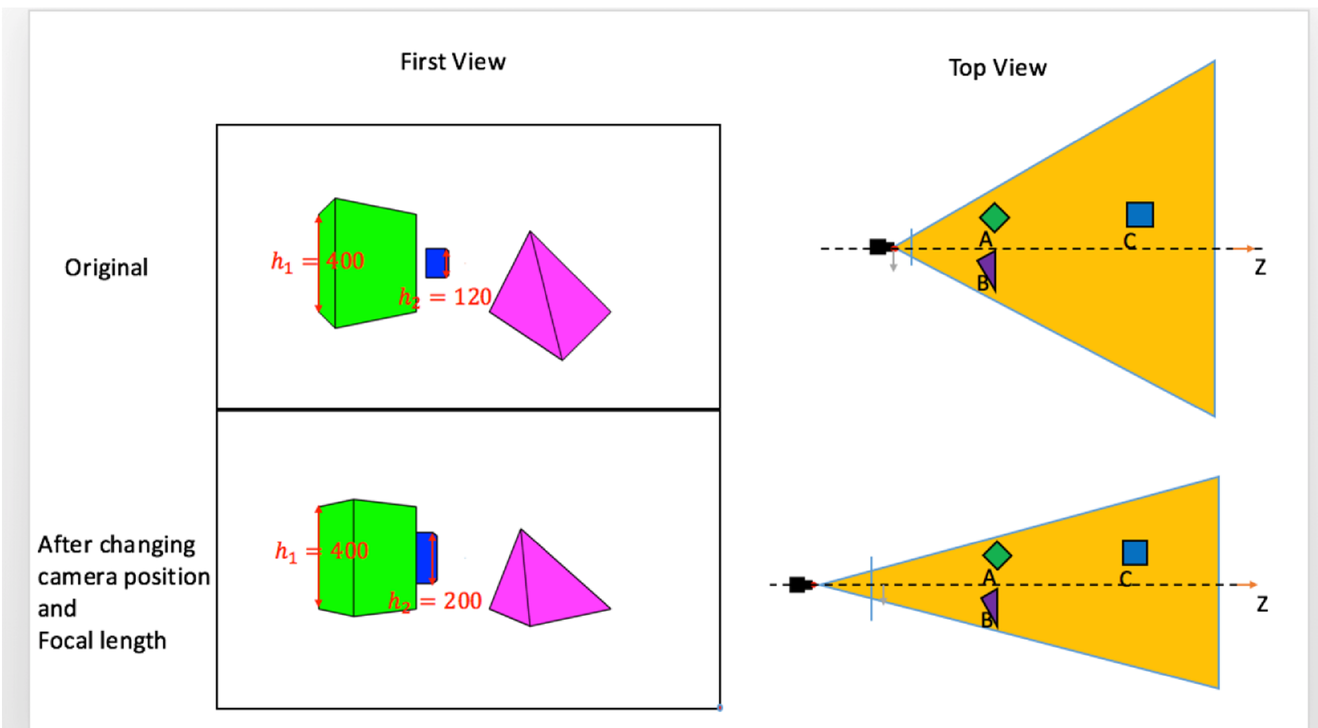
where (u, v) is the image coordinate of the projection, (X, Y, Z) is the 3D point, and f is the focal length of the camera. When the camera moves along with its Z-axis, the depth, Z, changes and therefore, the projection, (u, v), changes. In our paticular case the Z of interest is d_{ref}, the depth of the objects in the scene. In the following discussion we will only mention the u coordinate to simplify the equations, as we are focused mainly on height for the dolly zoom. This projection change produced by the depth change can be compensated by adjusting focal length:

$$u = f_{ref}\frac{X}{d_{ref}} = f'\frac{X}{d_{ref} - pos}$$

where pos is the movement of the camera along its Z axis ($+$ direction indicates approaching to objects) and f' is the modified focal length. f_{ref} and d_{ref} are the focal length and depth of an object in the original image, respectively. Dolly zoom effect exploits the compensation between depth and focal length, which produces depth sensation. The relationship between all the variable names as given in the code is described in the next figure, and when implementing the description in the code you should reference it.



Finally, the next figure illustrates the focal length/depth compensation: the camera moves away from the object while changing its focal length such that the height of the object A, $h_1 = 400$, in both original and moved images remains constant. Note that the heights of the other background objects are changed due to this effect.



3) Implementation

For the first part of this lab, you need to implement the function **compute_focal_length**. Given the depth of the object of interest d_{ref}, and the focal length f_{ref} of the camera for the original image, you need to estimate the modified focal length f', such that the height of the object remains constant as the camera moves in the Z-axis (different input pos values). When you have implemented the function, you can demonstrate the dolly soom effect in the synthetic scene above by using the **run_dolly_zoom** script we provide in the zip file Lab1.zip (http://courses.edx.org/asset-v1:PennX+ROBO2x+2T2017+type@asset+block@Lab1.zip).

Your Function

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
function [ f ] = compute_focal_length( d_ref, f_ref, pos )
% compute camera focal length using given camera position
%
% Input:
% - d_ref: 1 by 1 double, distance of the object whose size remains constant
% - f_ref: 1 by 1 double, previous camera focal length
% - pos: 1 by n, each element represent camera center position on the z axis.
% Output:
% - f: 1 by n, camera focal length

% Put your CODE here
f =

end
```

Code to call your function

Reset

```
d_ref = 4;
f_ref = 400;
pos = -5;
f = compute_focal_length( d_ref, f_ref, pos )
```

Run Function

Assessment:

Submit

Is the size of the output correct?

Is the estimated focal length correct?