# Assignment A2: Camera Functions

***CS 6320***
***Spring 2014***

**Assigned:** 6 January 2014

**Due:** 22 January 2014

For this problem, use handin to handin a pdf with the requested discussion and work (include name, date, assignment and class number in pdf), as well as the Matlab code. The code should conform to the style requested in the class materials.

**A1 Problem**: Develop a Matlab function for *Linear Matrix Equation of Perspective Projection* (Section 2.2.2, pp. 30-31, old text). (1) Validate the code on a hand computed set of points from the edges of a cube; i.e., discuss the test and predicted result and compare to the computed answer. (2) To help verify the code, show the results on sampled edge points from a cube for various values of the parameters; discuss results.

The function specification is:

```
function [u,v] = CS6320_intensity_camera(pts,R,t,alpha,beta,theta,u0,v0)
%
% CS5320_intensity_camera - simulate perspective camera
% On input:
%     pts (4xn array): x,y,z pts in scene in homogeneous coordinates
%        each column is an [x;y[z;1] point
%     R (3x3 rotation matrix): rotation matrix for camera frame in
%        world frame
%     t (3x1 vector): translation vector for camera in world frame
%     alpha (float): focal length times x scale parameter (f*k)
%     beta (float): focal length times y scale parameter (f*l)
%     theta (float): skew angle
%     u0 (float): pixel optical axis x value in image plane
%     v0 (float): pixel optical axis y value in image plane
% On output:
%     u (1xn vector): u values of imaged points
%     v (1xn vector): v values of imaged points
% Call:
%     cube = CS6320_gen_cube_pts;
%     T = Rot_k([0;0;1],0);
%     T(3,4) = -2;
%     [u,v] =
%        CS6320_intensity_camera(cube,T(1:3,1:3),T(1:3,4),1,1,pi/2,0,0);
% Author:
%     <Your name>
%     UU
%     Spring 2014
%
```