# CS 482 Homework 2 (Due September 21)

*J. Košecká*

1. **Perspective Projection** (5) It is often useful while testing some of the algorithms to simulate the perspective projection process. Write a MATLAB function, which implements the image formation process.
   Write a function $x = \texttt{project}(\texttt{X}, \texttt{R}, \texttt{T}, \texttt{K})$ which takes as an input image coordinates of 3D points in the world coordinate frame and generates pixel coordinates of the projected points in the image, assuming that $(R, T)$ is the displacement of the camera coordinate frame with respect to the world frame, $K$ is the matrix of intrinsic image parameters, and $X$ is a $3 \times n$ vector of the coordinates of 3D points. To test the function consider a unit cube placed in the origin of the world coordinate system (specified by 8 vertices $[0, 0, 0]', [1, 0, 0]', etc$, assume that the camera is translated along z-axis by some amount and rotated around x-axis by angle 20$\circ$. You can assume that matrix K is $[800, 0, 250; 0, 800, 250; 0, 0, 1]$. Generate the image of the cube. Its enough when you plot the vertices of the cube and optionally connect them by line to visualize it better. Submit the MATLAB code and generated MATLAB figure. It is commonly assumed that in the coordinate system of the camera the z-axis if pointing towards the scene and y down and x to the right.

2. **Vanishing Point** (5) Straight line in 3D world is projected in to a straight line in the image. The projections of two parallel line intersect in the image at so called *vanishing point*.

   a) Show (mathematically) that projections of parallel lines in the image intersect in a point.

   b) Compute for a certain family of parallel lines where in the image will the vanishing point be.

   c) When does the vanishing point of the lines in the image lie at infinity (i.e. the lines in the image do not intersect)?

3. **Convolution** (2.5) Given an image http://cs.gmu.edu/~kosecka/cs482/rose.jpg generate a smoothed image of the same size, which is obtained by convolution with a box filter of varied size (3x3 7x7, 11x11,1x7 and 7x1). Visualize the results and comment on the effect of the size of the filter on the final result. Post the code and the images on the web site (you can use the build in convolution functions of MATLAB and OpenCV). What is the complexity (in terms of number of operations, additions and multiplications) of the convolution operation as a function of image size and filter size ?

4. **Smoothing** (2.5) Write a function in MATLAB that takes two arguments, a width parameter, $w$, and a variance parameter, $\sigma$ and returns a 1D array containing a Gaussian filter (mask) of the desired dimension and variance. Optionally you can normalize the kernel such that the sum of all the elements in the array is 1. Use this function and the conv2 routine to convolve the images with a Gaussian filter of size 5 and sigma of 1. Repeat with a Gaussian kernel of size 11 with a sigma of 3. Comment on the difference between the two images. Assume that the relationship between size of the filter and $\sigma$ is $6\sigma - 1 = n$. Convolution of an image with 1D filter in x and y directions can be accomplished by the following call of the function `conv2`, where `filter` is a 1D filter.
   ```
   fsmooth = conv2( conv2( f, filter', 'same' ), filter, 'same' );
   ```
   Hand in MATLAB code gaussian.m and post two images obtained after filtering.