# CS 532 – Computer Vision
# Assignment 1

Akshay Atam
aatam@stevens.edu

## Problem 1

This task involved using Direct Linear Transformation (DLT) to warp an image of a basketball court. Inverse warping was used to fill the pixels and bilinear interpolation was used to fill in the gaps. A total of four points were chosen in the input image – [0,200], [246,45], [419,69], [287,318]. These points were mapped onto the output image, a 940 by 500 image. The pixels chosen has been shown in the jupyter notebook. The next step was to generate the homography from those pixel points. Class slides and Zisserman's book were used to get the knowledge.

For extra credits, we were to use normalization to normalize the points based on the dimensions of image. However, in the book, I also saw the use of mean and standard deviation to scale the points. I implemented both of the techniques and generated the homography for the two approaches. While the homography was generated successfully, the warping could not be done as I was encountering certain difficulties.

For implementing DLT using line feature locations, the points chosen earlier were first converted to homogenous coordinates (by adding a 1) and doing cross product for the four points to get four-line equations. These line equations were of the form (x,y,w) and inhomogeneous homography approach was used to get the homography. Since there are a total of 4 lines and each line gives 2 equations, we will get a total of 8 equations for the homography. The homography has been shown as an output.

## Problem 2

This task involved using object centered motion to fix the fish in the image and design a path that completes a half circle. The sample MATLAB file was looked upon and a basic dolly zoom effect was created where the background was moving away from the foreground (fish). To generate the rotation effect, I noticed that by manipulating the rotation matrix with the step size for the loop, we could achieve the desired result. The angle for the rotation matrix was manipulated and for simplicity, only rotation in the z-axis was performed. The rotations can be performed in the x and y axes as well, generating a unique form of rotation. A half circle was easier to see in the z axis.

The loop was run for 80 iterations with each iteration getting a rotation according to the step size. The final video was then made from these images (I have referred to link I used to get the knowledge to get video from images in the MATLAB file). The final video shows the image moving back and rotating counter-clockwise to get a half circle effect.