

# CMPT 742: Practices in visual computing I

## Assignment 3

### Epipolar geometry

Total points: 30

---

#### Introduction

In this assignment we want to use epipolar geometry to find the structure of a 3D scene given only two images of different views of the scene. More specifically, given two images of a scene we want to find the **Fundamental matrix** between the two images. For this, we will use the 8-point algorithm which you have already seen in class.

#### Normalized 8-point algorithm (10 points)

In the file `eight_point_fw` Implement the function `find_fundamental_matrix`. This function takes two sets of corresponding keypoints from the two images, and computes the fundamental matrix. For stability of the algorithm, we would want to first normalize the points. For this we follow the following steps:

1. Find the centroid of the points (find the mean x and mean y value)
2. Compute the mean distance of all points to the centroid
3. Construct a transformation matrix that transforms the centroid to the origin and makes the mean distance of the points to the centroid equal to  $\sqrt{2}$ .

You can use the matrix to normalize the points, find the fundamental matrix on the normalized points, and then adjust the matrix to the denormalized points. You can read more about this method [here](#).

#### RANSAC (10 points)

The keypoint detection algorithm that we use as a preprocessing step contains errors and inaccuracies, and therefore could affect the final result of our 8-point algorithm. For a more robust result you can use the RANSAC algorithm. You should follow these steps and complete the function `find_fundamental_matrix_ransac`:

1. Sample 8 points from the set of corresponding points.
2. Compute the fundamental matrix using those 8 points.
3. Find the number of inliers with the current fundamental matrix.
4. repeat from step 1 until a certain number of iterations has been reached.
5. return the best fundamental matrix.

## Report (10 points)

You should include a report that contains the fundamental matrices you computed for different examples, the output images and what these output images mean.

To check your result, You can compare your fundamental matrix with the OpenCV built-in function. For the first part, Your result should be exactly the same as the output of the built-in function (except for very small differences caused by floating point computations). For the RANSAC part, your result might differ from the built-in function's output. However, your final images should make sense.