EEE 515 **HOMEWORK ASSIGNMENT 3**

Due on Monday, October 30th, 2023 at 11:59 pm

1 Written Problems

Problem 1. *(Homogenous Coordinates, Points, and Lines*

1. Show that the line $y = mx + b$ can be written in homogenous coordinates as the following: $\mathbf{l}^T\mathbf{x} = 0$, and show the parameters for homogenous line $\mathbf{l}$ and homogenous point $\mathbf{x}$.
2. Give at least two homogeneous representations (x, y, z) for the point $(3, 5)$ on the 2D plane, one with $z > 0$ and one with $z < 0$.
3. Take two homogenous lines, $\mathbf{l}_1, \mathbf{l}_2$. Show that $\mathbf{l}_1 \times \mathbf{l}_2$ is the point of intersection between the two lines.
4. Consider the two lines $x+y-5 = 0$ and $4x-5y+7 = 0$. Compute the point of intersection in homogenous space, and then convert it into standard Cartesian space.
5. Take a line with standard form $ax + by + c = 0$. Write the general equation for the intersection point for any parallel line to this line. Where does this point lie in Cartesian space?
6. Consider two homogenous points $\mathbf{x}_1, \mathbf{x}_2$. Write (and show justification) an expression for the homogenous line $\mathbf{l}$ that goes through both points.

Problem 2. *(2D transformations)*

1. Derive a single homogenous $3x3$ matrix for rotation of angle θ around a point (a, b). You must show steps to justify the derivation (not merely giving the answer).
2. A square has vertices $p_1 = (1, 1)$, $p_2 = (2, 1)$, $p_3 = (2, 2)$ and $p_4 = (1, 2)$. Calculate the new vertices of the square after a rotation about p_2 through an angle of 45 degrees.
3. Construct the 2D transformation (one matrix) to reflect across an arbitrary line $ax+by+c = 0$. [Hint: First consider $b \neq 0$. One should translate and rotate to align the line with the x-axis, reflect across the x-axis, and then restore the line back to its normal place. The final matrix should be only in terms of a, b, c and θ as the angle of the line. Show that this matrix also satisfies the special case when $b = 0$ as well.

Problem 3. *(Affine + second order warp)* Consider an image transform that maps (x, y) to (x', y') via the following equations:

$$x' = ax + by + t_x + \alpha x^2 + \beta y^2, y' = cx + dy + t_y + \gamma x^2 + \theta y^2$$

.

Given two images with point-correspondences (x, y) to (x', y'), derive a technique to solve all parameters in closed-form. What is the minimal number of points needed to solve this? Note: you don't have to actually solve this problem, just setup the way to solve it.

2 Coding Assignments

Problem 4. *2D Transformations on Images*

1. Construct an image of 300×300 pixels, with a white irregular quadrilateral (i.e. four unique corners) approximately size 50×50 pixels is centered in the picture on a black background. Perform the following operations: (1) Translate the image by $(30, 100)$ and (2) Rotate the image by 45 degrees using OpenCV commands about the original center. Display the rotated quadrilateral.
2. Come up with a feature detector and descriptor to accurately match the four corners of the quadrilateral in the two pixels. **You cannot use any built-in feature detectors/descriptors to do this step.** This could include implementing Harris Corners from scratch and coming up with your own descriptor. However, there might be simpler features that will give you the matching you need.
3. Once you have established point correspondences with your feature detector/descriptor, pick a subset of them (how many do you need?), and then construct a linear least squares optimization to solve for θ and t_x, t_y. You should recover back the translation and rotation in the first part.

Problem 5. *Image Stitching*

1. Capture at least 4 images to stitch together of a suitable scene in Arizona or wherever you live using your phone camera. Display the images you captured in the report.
2. Choose one image as the reference image. Choosing the middle images of the sequence as the reference frame is the preferred option, as it will result in mosaics with less distortion. However, you can choose any image of the sequence as the reference frame. If you choose end images as the reference frame, directly estimating homography between these images will be difficult as they have very small overlap. To deal with this issue, you might have to compose homographies together.
3. **Feature Matching:** Match feature descriptors between two images. That is, you will need to find pairs of features that look similar and are thus likely to be in correspondence. We will call this set of matched features "tentative" correspondences. As the first step, use Euclidean distance to compute pairwise distances between the SIFT descriptors. Then, you need to perform "thresholding". For thresholding, use the simpler approach due to Lowe

of thresholding on the ratio between the first and the second nearest neighbors. This is described in Section 5 in "Multi-Image Matching using Multi-Scale Oriented Patches" by Brown et al. . `http://matthewalunbrown.com/papers/cvpr05.pdf`. Consult Figure 6b in the paper for picking the threshold. Note: You can ignore the fast indexing described in section 6 of the paper. You can visualize the tentative correspondences between two images by displaying the feature displacements (choose a reasonable number to display on the images so that it is clear whether the matching is working or not).

4. **Homography estimation with RANSAC:** Compute the homography estimation for the images with RANSAC. You are allowed to use openCV functions to do so, but please document the functions and parameters used to estimate homographies. You must write down the homography matrices in the report as well as report the reprojection error on your features for each matrix.

5. **Warping and compositing:** Warp each image into the reference frame using the estimated homographies and composite warped images into a single mosaic. In Python, you can make use of the OpenCV function cv2.warpPerspective. Display the final homography.