CS 415 Mini Project 4

1 Question Answering

Q1. What is the difference between invariance and equivariance? Is the Harris corner detector invariant or equivariant to each of the image transformations below? Please briefly justify your answer. (15 points)

- Translation
- Rotation
- Horizontal flipping
- Scaling
- Adding a constant to every pixel intensity (ignoring overflow)

Q2. When designing feature descriptors such as SIFT, what is the benefit of using image gradients, histogram, and cells? (10 points)

Q3. For each of the desired transformations below, provide the 3x3 transformation matrix that could be used to transform an arbitrary point in the homogeneous coordinate. Suppose the top-left corner of an image is the origin, and the x and y axes point to the right and bottom, respectively. Hint: in this coordinate system, the Θ in the rotation matrix defined in the slides will correspond to the clockwise direction. (15 points)

- Shift to the right by 100 pixels
- Rotate around the origin in the clockwise direction by 45 degrees
- Rotate around the point (20, 20) in the counterclockwise direction by 90 degrees

2. Programming

P1. Follow the steps below to apply geometric transformations to pixel coordinates.

- Create an all-white (RGB) image of a resolution 500x500
- Use the cv2.circle() method[1] to draw a solid red circle with a radius 10 at ***a***=(100, 40). 100 and 40 are respectively the x-coordinate and y-coordinate, using the same coordinate system as in Q3. (5 points)
- Rotate ***a*** around the origin in the clockwise direction by 60 degrees to get a new point ***b***. Implement this rotation by first defining/calculating the transformation matrix and then applying it via matrix multiplication. Draw a solid green circle at ***b***. (10 points)
- Draw a solid black circle at ***c*** = (100, 100). Rotate ***a*** around ***c*** in the clockwise direction by 60 degrees to get a new point ***d***. Draw a solid blue circle at ***d***. (10 points)

[1] https://www.geeksforgeeks.org/python-opencv-cv2-circle-method/

P2. Self-study the cv2.warpAffine() method[2], which can apply any affine transformation to an image. For each of the transformations below, independently apply it to the **original image** of lena.png via cv2.warpAffine(). Keep the image size unchanged. You need to define and calculate the transformation matrix by yourself without calling any existing functions for this purpose. It is expected that a portion of the transformed image may go out of the original coordinate space and be cut off.

- Move the image to the right by 100 pixels and to the bottom by 200 pixels. (5 points)
- Flip the image horizontally with respect to the image center. (5 points)
- Rotate the image around the origin in the clockwise direction by 45 degrees. (10 points)
- Rotate the image around the image center in the clockwise direction by 45 degrees. (15 points)

3 Submission

Please follow the instructions below for submission.

- You need to upload two files to Blackboard: a PDF file and a .py file[3]. Do not compress them into a single ZIP file.
- The PDF file contains all your solutions to this homework. For Question Answering, you can either type answers or handwrite them and take a photo. For Programming, you need to include output of the program such as a processed image.
- The .py file contains all your code for the programming problems.

[2] https://www.geeksforgeeks.org/python-opencv-affine-transformation/

[3] Using Jupyter Notebook and submitting a .ipynb file instead of a .py file are fine.