Intro to Image Understanding (CSC420)

Assignment 4

Posted: March 18, 2024 **Submission Deadline : March 27, 11.59pm, 2024**

Max points: 15

1. **[3 points]** In this exercise, you are asked to take a photo. In particular, please take a planar item/object for which you know the real-world width and height (in cm), for example a piece of paper or a dollar bill. Tape the item on the door. Take a picture of the door such that all four corners of the door are visible on the photo. Take this picture in an oblique view, ie, the door is not a perfect rectangle but rather a quadrilateral in the photo. Estimate the width and height of the door (in cm) from the picture.

2. **[5 points]** You are given a few photos of landscape. The goal is to take two photos, LANDSCAPE_1 and LANDSCAPE_2 and stitch them into one photograph. You can do this by extracting SIFT features from both photos, match them, and estimate a homography of one photo with respect to the other. Use RANSAC to find the best homography. Once you compute the homography, "stitch" the two photos together, forming a small panorama. We will give half points if you compute affine transformation instead of a homography.

3. Attached is an image UM_000038.PNG recorded with a camera mounted on a car. The focal length of the camera is 721.5, and the principal point is $(609.6, 172.9)$. We know that the camera was attached to the car at a distance of 1.7 meters above ground.

   (a) **[1 point]** Write the internal camera parameter matrix K.

   (b) **[1 point]** Write the equation of the ground plane in camera's coordinate system. You can assume that the camera's image plane is orthogonal to the ground.

   (c) **[2 points]** How would you compute the 3D location of a 2D point (x, y) in the image by assuming that the point lies on the ground? You can assume that the camera's image plane is orthogonal to the ground. No need to write code, write a mathematical explanation.

4. **[3 points]** Describe (in mathematical form, no code) how to compute disparity for a pair of parallel stereo cameras. Please include an algorithm (pseudo-code) that includes mathematical details. What is the computational complexity of this algorithm? How do you compute depth for each pixel?