Intro to Image Understanding (CSC420)
Assignment 4

Posted: Nov 16, 2019 **Submission Deadline : Nov 24, 11.59pm, 2019**

Max points: 15

1. 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.

2. In this exercise you are given stereo pairs of images from the autonomous driving dataset KITTI (`http://www.cvlibs.net/datasets/kitti/index.php`). The images have been recorded with a car driving on the road. Your goal is to create a simple system that analyzes the road ahead of the driving car. Include your code to your solution document.
   - In your data directory you are given a folder called TRAIN and TEST. You will need to compute all your results only for the TEST folder. Using TRAIN is optional. For all parts, you only need to process the **first three images** written in DATA/TEST/TEST.TXT.

   (a) [**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?
   (b) [**3 points**] What is a fundamental matrix? Describe an algorithm (pseudo-code) to compute the fundamental matrix from a pair of uncalibrated (non-parallel) cameras. Include mathematical details. What is the computational complexity of the algorithm? Is it possible to compute depth for a pair of images taken from non-parallel cameras for which you do not know the relative pose/distance? Please provide an argument for your answer.
   (c) [**1 point**] The folder results under TEST contains stereo results from the algorithm called SPSSTEREO. For each image there is a file with extension LEFT_DISPARITY.PNG that contains the estimated disparity.
   For each image compute depth. In particular, compute depth which is a $n \times m$ matrix, where n is the height and m the width of the original image. The value $depth(i, j)$ should be the depth of the pixel (i, j). In your solution document, include a visualization of the depth matrices. In order to compute depth you'll need the camera parameters:
   In the TEST/CALIB folder there are text files with extension ALLCALIB.TXT which contain all camera parameters. Note that the baseline is given in meters.

- The folder results under test contains detections in a file DETS-TEST.ZIP, which has detection results for each image. The variable DS contains a detection in each row of the matrix. The results are in Matlab (.MAT) format. Python users can use the SCIPY.IO.LOADMAT function to lead this data. Each row represents a rectangle (called a bounding box) around what the detector thinks it's an object. The bounding box is represented with two corner points, the top left corner (x_{left}, y_{top}) and the bottom right corner (x_{right}, y_{right}). Each row of DS has the following information: $[x_{left}, y_{top}, x_{right}, y_{bottom}, id, score]$. Here SCORE is the confidence of the detection, i.e., it reflects how much a detector believes there is an object in that location. The higher the better. The variable ID reflects the viewpoint of the detection and you can ignore it for this assignment.

(d) [**1 points**] In your solution document, include a visualization of the first three images with all the car, person and cyclist detections. Mark the car detections with red, person with blue and cyclist with cyan rectangles. Inside each rectangle (preferably the top left corner) also write the label, be it a car, person or cyclist.

(e) [**3 points**] Compute the 3D location of each detected object. How will you do that? Come up with an approach to get a (2D) segmentation of each object. Can you use depth to help you?