Assignment 1: CS 763, Computer Vision

Due: 27th Jan before 11:00 pm

Remember the honor code while submitting this (and every other) assignment. All members of the group should work on and understand all parts of the assignment. We will adopt a zero-tolerance policy against any violation.

Submission instructions: You should ideally type out all the answers in Word (with the equation editor) or using Latex. In either case, prepare a pdf file. For assignment submission, follow the instructions for arrangement of folders and subfolders as given in `https://drive.google.com/open?id=1ltPhTIWmpUOqWgkZAVmCA_svNKexgRjw`. Create a single zip or rar file obeying the aforementioned structure and name it as follows: A1-IdNumberOfFirstStudent-IdNumberOfSecondStudent-IdNumberOfThirdStudent.zip. (If you are doing the assignment alone, the name of the zip file is A1-IdNumber.zip). Upload the file on moodle BEFORE 11:00 pm on 27th January. Late assignments will be assessed a penalty of 50% per day late. Note that only one student per group should upload their work on moodle. Please preserve a copy of all your work until the end of the semester. If you have difficulties, please do not hesitate to contact.

1. In this question, you are required to change the pose of a base human skeleton given the joint angles of all the joints. Consider the skeleton code provided in the file `myScript.m`. It includes the information about base skeleton, kinematic chain and two test cases. You are required to implement two functions, a) `angle2rot()` which returns the rotation matrix for given angles around X, Y and Z axes, and b) `transformPose()` that transforms the base pose as per the given joint rotation matrix, base pose co-ordinates, kinematic chain and the index of the root joint, and returns the new pose. More details about the variables is provided in the code. Note that the provided joint angles are with respect to the parent's frame of reference. The code also includes the function `plotPose()` for visualizing your outputs. Your code should be generic and not hard coded for the test cases provided.

2. In this question, we will perform distortion correction. Consider the images provided in the input folder. The file `code/myScript.m` provides the code for image distortion and calls routine `radUnDist()` to perform undistortion. The images have been distorted using the following distortion model and distortion parameters $\mathbf{q} = \{q_1, q_2\}$:
$$\mathbf{x_d} = \mathbf{x_u}(1 + q_1 r + q_2 r^2)$$
Your task is to implement the function `radUnDist()` that undistorts the distorted image given the distortion parameters. You need to iteratively find x_u until convergence (should converge quickly): Once done, warp the image intensities at $\mathbf{x_d}$ to $\mathbf{x_u}$ using MATLAB's interp2 function. Display the undistorted image and include the code in the submission. **[20 points]**

3. Given a projective transformation matrix:
$$M = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$
find the image of a hyperbola $y = 1/x$ under the transform M. (Represent the hyperbola in parametric form $(t, 1/t), t \neq 0$ in homogeneous coordinates and find the projection.) Comment on the shape of the image of hyperbola obtained. **[10 points]**

4. Here you will develop an application of the concept of vanishing points to camera calibration. Consider two images (I_P and I_Q) of a non-planar scene taken with two pinhole cameras having unknown focal lengths f_p and f_q respectively. Both cameras produce images on a Cartesian grid with aspect ratio of 1 and unknown resolution s_p and s_q respectively. The orientations and positions of the two cameras are related by an unknown rotation (given by a 3×3 rotation matrix $\mathbf{R}$) and an unknown translation (given by a 1×3 vector $\mathbf{t}$). Note that 'position' here refers to the location of the camera pinhole, and 'orientation' refers to the XYZ axes of the camera coordinate system. In both I_P and I_Q, suppose you accurately mark out the corresponding vanishing points of three mutually perpendicular directions ℓ_1, ℓ_2, ℓ_3 in the scene. (Obviously, all three directions were visible from both cameras). Let the vanishing points have coordinates (p_{1x}, p_{1y}), (p_{2x}, p_{2y}) and (p_{3x}, p_{3y}) in I_P, and (q_{1x}, q_{1y}), (q_{2x}, q_{2y}) and (q_{3x}, q_{3y}) in I_Q. Note that these coordinates are in terms of pixel units and the correspondences are known. Given all this information, can you infer $\mathbf{R}$? Can you infer $\mathbf{t}$? Can you infer f_p and f_q? Can you infer s_p and s_q? Explain how (or why not). (Hint: You can start off by assuming that you knew all the intrinsic parameters and work yourself upwards from there). **[20 points]**

5. a) Prove (algebraically) that the projections (in image plane) of any two parallel lines L_1, L_2 in $\mathbb{R}^3$ have an intersection point, the vanishing point.
   b) Prove (algebraically) that the vanishing points corresponding to three (different) sets of parallel lines on a 3D plane are collinear in the image plane. **[5 + 5 points]**

6. **Image Forensics:** Consider the image painting.jpg provided in the input directory of Question 6. It is the image of the painting *Flagellation of Christ* by Piero della Francesca. Assume that the green line on the image is the horizon line and the also that the height of Christ, **R** is 180cm. The painting has been lauded for its correct depiction of linear perspectives. Using your knowledge of Computer Vision, can you find the height, **H**, of the person on the right? Mention the estimated height in the report along with a detailed description of how you arrived at the answer. **[10 points]**