Intro to Image Understanding (CSC420)
Assignment 4

Due Date: Mar 31st, 2023, 11:59 pm
Total: 160 marks

General Instructions:

- You are allowed to work directly with one other person to discuss the questions. However, the implementation and the report should be your own original work; i.e. you should not submit identical documents or codes. If you choose to work with someone else, write your teammate's name on top of the first page of the report.
- Your submission should be in the form of an electronic report (PDF), with the answers to the specific questions (each question separately), and a presentation and discussion of your results. For this, please submit a file called **report.pdf** to MarkUs directly.
- Submit documented codes that you have written to generate your results separately. Please store all of those files in a folder called **assignment4**, zip the folder, and then submit the file **assignment4.zip** to MarkUs. You should include a **README.txt** file (inside the folder) which details how to run the submitted codes.
- Do not worry if you realize you made a mistake after submitting your zip file; you can submit multiple times on MarkUs.
- MarkUs has a file size limit. If your pdf or zip file is larger than the limit, you can try resizing or reducing the resolution of images in your report to reduce the file size. If that does not work, you can split your report into multiple files (e.g. Report_part_1_of_3.pdf, Report_part_2_of_3.pdf, etc.)

Part I: Theoretical Problems (70 marks)

[Question 1] RANSAC (10 marks)

We have two images of a planar object (e.g. a painting) taken from different viewpoints and we want to align them. We have used SIFT to find a large number of point correspondences between the two images and visually estimate that at least 70% of these matches are correct with only small potential inaccuracies. We want to find the true transformation between the two images with a probability greater than 99.5%.

1. **(5 marks)** Calculate the number of iterations needed for fitting a homography.
2. **(5 marks)** Without calculating, briefly explain whether you think fitting an affine transformation would require fewer or more RANSAC iterations and why.

[Question 2] Camera Models (30 marks)

Assume a plane passing through point $\vec{P_0} = [X_0,\ Y_0,\ Z_0]^T$ with normal $\vec{n}$. The corresponding vanishing points for all the lines lying on this plane form a line called the horizon. In this question, you are asked to prove the existence of the horizon line by following the steps below:

1. **(15 marks)** Find the pixel coordinates of the vanishing point corresponding to a line L, passing point $\vec{P_0}$ and going along direction $\vec{d}$.

   Hint: $\vec{P} = \vec{P_0} + t\,\vec{d}$ are the points on line L, and $\vec{p} = \begin{pmatrix} \omega x \\ \omega y \\ \omega \end{pmatrix} = K\,\vec{P} = K \begin{pmatrix} X_0 + t\,d_x \\ Y_0 + t\,d_y \\ Z_0 + t\,d_z \end{pmatrix}$ are pixel coordinates of the same line in the image, and $K = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}$, where f is the camera focal length and (p_x, p_y) is the principal point.

2. **(15 marks)** Prove the vanishing points of all the lines lying on the plane form a line.

   Hint: all the lines on the plane are perpendicular to the plane's normal $\vec{n}$; that is, $\vec{n}\,.\,\vec{d} = 0$, or $n_x\,d_x + n_y\,d_y + n_z\,d_z = 0$

[Question 3] Homogeneous Coordinates (30 marks)

Using the homogeneous coordinates:

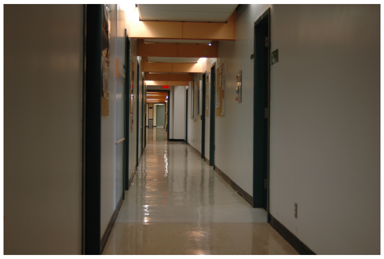
1. **(15 marks)** (a) Show that the intersection of the 2D line l and l' is the 2D point $p = l \times l'$.
   (here × denotes the cross product)

2. **(15 marks)** (b) Show that the line that goes through the 2D points p and p' is $l = p \times p'$.

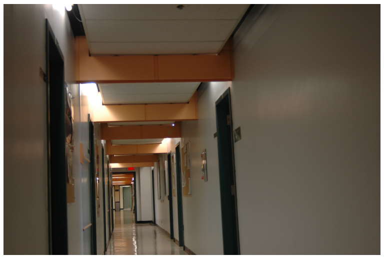
Part II: Implementation Tasks (90 marks)

[Question 4] Homography (60 marks)

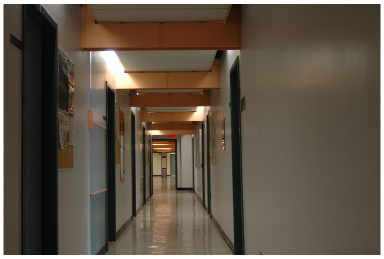
You are given three images `hallway1.jpg`, `hallway2.jpg`, `hallway3.jpg` which were shot with the same camera (i.e. same internal camera parameters), but held at slightly different positions/orientations (i.e. with different external parameters).



`hallway1.jpg`



`hallway2.jpg`



`hallway3.jpg`

Consider the homographies **H**,

$$\begin{bmatrix} \widetilde{w}\widetilde{x} \\ \widetilde{w}\widetilde{y} \\ \widetilde{w} \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

that map corresponding points of one image I to a second image $\widetilde{I}$, for three cases:

A. The right wall of I =`hallway1.jpg` **to the right wall of** $\widetilde{I}$ =`hallway2.jpg`.
B. The right wall of I =`hallway1.jpg` **to the right wall of** $\widetilde{I}$ =`hallway3.jpg`.
C. The floor of $\widetilde{I}$ =`hallway1.jpg` **to the floor of** $\widetilde{I}$ =`hallway3.jpg`.

For each of these three cases:

1. **(10 marks)** Use a Data Cursor to select corresponding points by hand. Select more than four pairs of points. (Four pairs will give a good fit *for those points*, but may give a poor fit for other points.) Also, avoid choosing three (or more) collinear points, since these do not provide independent information. This is trickier for case **C**. Make two **figures** showing the gray-level images of I and $\widetilde{I}$ with a colored square marking each of the selected points. You can convert the image I or $\widetilde{I}$ to gray level using an RGB to grayscale function (or the formula $gray = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$).

2. **(10 marks)** Fit a homography **H** to the selected points. Include the estimated **H** in the report, and describe its effect using words such as *scale, shear, rotate, translate*, if appropriate. You are not allowed to use any homography estimation function in OpenCV or other similar packages.

3. **(10 marks)** Make a **figure** showing the $\widetilde{I}$ image with red squares that mark each of the selected $(\widetilde{x}, \widetilde{y})$, and green squares that mark the locations of the estimated $(\widetilde{x}, \widetilde{y})$, that is, use the homography to map the selected (x, y) to the $(\widetilde{x}, \widetilde{y})$ space.

4. **(25 marks)** Make a figure showing a new image that is larger than the original one(s). The new image should be large enough that it contains the pixels of the I image as a subset, along with *all* the inverse mapped pixels of the $\widetilde{I}$ image. The new image should be constructed as follows:

   - RGB values are initialized to zero,
   - The red channel of the new image must contain the `rgb2gray` values of the I image (for the appropriate pixel subset only);
   - The blue and green channels of the new image must contain the `rgb2gray` values of the corresponding pixels $(\widetilde{x}, \widetilde{y})$ of $\widetilde{I}$. The correspondence is computed as follows: for each pixel (x, y) in the new image, use the homography $\mathbf{H}$ to map this pixel to the $(\widetilde{x}, \widetilde{y})$ domain (not forgetting to divide by the homogeneous coordinate), and round the value so you get an integer grid location. If this $(\widetilde{x}, \widetilde{y})$ location indeed lies within the domain of the $\widetilde{I}$ image, then copy the `rgb2gray`'ed value from that $\widetilde{I}(\widetilde{x}, \widetilde{y})$ into the blue and green channel of pixel (x, y) in the new image. (This amounts to an inverse mapping.)
     If the homography is correct and *if the surface were Lambertian** then corresponding points in the new image would have the same values of R,G, and B and so the new image would appear to be gray at these pixels.
   - Based on your results, what can you conclude about the relative 3D positions and orientations of the camera? Give only qualitative answers here. Also, What can you conclude about the surface reflectance of the right wall and floor, namely are they more or less Lambertian? Limit your discussion to a few sentences.

(5 marks) Along with your writeup, hand in the program that you used to solve the problem. You should have a switch statement that chooses between cases **A, B, C**.

* *Lambertian reflectance is the property that defines an ideal "matte" or diffusely reflecting surface. The apparent brightness of a Lambertian surface to an observer is the same regardless of the observer's angle of view. Unfinished wood exhibits roughly Lambertian reflectance, but wood finished with a glossy coat of polyurethane does not, since the glossy coating creates specular highlights. Specular reflection, or regular reflection, is the mirror-like reflection of waves, such as light, from a surface. Reflections on still water are an example of specular reflection.*

[Question 5] Mean Shift Tracking (30 marks)

In tutorial 10, we learned about the mean shift and cam shift tracking. In this question, we first attempt to evaluate the performance of mean shift tracking in a single case and will then implement a small variation of the standard mean shift tracking. For both parts you can use the attached short video `KylianMbappe.mp4` or, alternatively, you can record and use a short (2-3 second) video of yourself. You can use any OpenCV (or other) functions you want in this question.

1. **(20 marks)** Performance Evaluation

   - Use the Viola-Jones face detector to detect the face on the first frame of the video. The default detector can detect the face in the first frame of the attached video. If you record a video of yourself, make sure your face is visible and facing the camera in the first frame (and throughout the video) so the detector can detect your face in the first frame.
   - Construct the **hue** histogram of the detected face on the first frame using appropriate **saturation** and **value** thresholds for masking. Use the constructed **hue** histogram and mean shift tracking to track the bounding box of the face over the length of the video (from frame #2 until the last frame). So far, this is similar to what we did in the tutorial.
   - Also, use the Viola-Jones face detector to detect the bounding box of the face in each video frame (from frame #2 until the last frame).
   - Calculate the intersection over union (IoU) between the tracked bounding box and the Viola-Jones detected box in each frame. Plot the IoU over time. The x axis of the plot should be the frame number (from 2 until the last frame) and the y axis should be the IoU on that frame.
   - In your report, include a sample frame in which the IoU is large (e.g. over 50%) and another sample frame in which the IoU is low (e.g. below 10%). Draw the tracked and detected bounding boxes in each frame using different colors (and indicate which is which).
   - Report the percentage of frames in which the IoU is larger than 50%.
   - Look at the detected and tracked boxes at frames in which the IoU is small ($< 10\%$) and report which (Viola-Jones detection or tracked bounding box) is correct more often (we don't need a number, just eyeball it). Very briefly (1-2 sentences) explain why that might be.

2. **(10 marks)** Implement a Simple Variation

   - In the examples in Tutorial 10 (and the previous part of this question) we used a **hue** histogram for mean shift tracking. Here, we implement an alternative in which a histogram of gradient direction values is used instead.
   - After converting to grayscale, use blurring and the Sobel operator to first generate image gradients in the x and y directions (I_x and I_y). You can then use `cartToPolar` (with `angleInDegrees=True`) to get the gradient magnitude and angle at each frame. You can use 24 histogram bins and [0,360] (i.e. not [0,180]) directions.
   - When constructing **hue** histograms, we thresholded **saturation** and **value** channels to create a mask. Here, you can threshold the gradient magnitude to create a mask. For example, you can mask out pixels in the region of interest in which the gradient magnitude is less than 10% of the maximum gradient magnitude in the RoI.

- Calculate the intersection over union (IoU) between the tracked bounding box and the Viola-Jones detected box in each frame. Plot the IoU over time. The x axis of the plot should be the frame number (from 2 until the last frame) and the y axis should be the IoU on that frame.
- In your report, include a sample frame in which the IoU is large (e.g. over 80%) and another sample frame in which the IoU is low (e.g. below 50%). Draw the tracked and detected bounding boxes in each frame using different colors (and indicate which is which).
- Report the percentage of frames in which the IoU is larger than 50%.