ECE661 Fall 2024: Homework 3

TA: Rahul Deshmukh (deshmuk5@purdue.edu)

Due Date: Midnight, 11 Sep 2024

Late submissions will be accepted with penalty: -10 points per-late-day, up to 5 days.

In this homework, you will explore and compare different approaches for removing the perspective and affine distortions in images. Note that you can NOT use any built-in functions to compute homography matrices or for image warping.

Turn in typed solutions via BrightSpace. Additional instructions can be found at BrightSpace. Your submission should be two files – (1) Homework PDF (2) ZIP file containing source code (only *.py and *.txt files)

1 Introduction

The goal of this homework is to remove projective and affine distortions in the given images. This is referred as metric rectification in the text [1]. For your programming tasks, you will implement the approaches given the textbook for metric rectification. Make sure to read the description for the given tasks carefully. This homework is based on the concepts covered in Lectures 4 and 5.

You will show results after the distortions removal on the given input images as well as using your own images for the three approaches detailed in the rest of this section. Note that the set of points or lines you pick for estimating any homography should reside on the same planar surface in the scene.

1.1 Point-to-point correspondences

This approach is a trivial extension of what you implemented in your Homework 2. You will manually find point-to-point correspondences between the original undistorted scene as your domain and its photograph as your range which has projective and affine distortions. The inverse homography will eliminate these distortions. Obviously, using point-to-point correspondences is the most straightforward method compared to the other two. However, it often requires a large number of correspondences to give a numerically stable solution for estimating the homography.

1.2 The Two-Step Method

In the two-step method, you first remove the projective distortion using the Vanishing Line (VL) method discussed in Lecture 4. Subsequently, you remove the affine distortion by using the $cos(\theta)$ expression with θ equal to 90° . Note that you must first remove the projective distortion before you can remove the affine distortion with the $cos(\theta)$ based method. In the text [1], the two-step approach is termed as **stratified**.

Removing the Projective Distortion: The projective distortion can be eliminated by the homography that takes the VL back to $\boldsymbol{l}_\infty$. To do so, you first estimate a VL in the image plane by picking pixel coordinates of at least two pairs of parallel lines in the original scene. Taking the cross-product of two such pixels on any line in the image will give you the HC representation of that line. Taking the cross-product of the 3-vectors for two different lines (which are parallel in the original scene) will give you the HC representation for the Vanishing Point (VP) for those two lines. Then taking the cross-product of two such VPs for two different pairs of parallel lines will give you the VL you need for getting rid of the projective distortion.

Removing the Affine Distortion: The affine distortion can be eliminated by the homography that restores the angle between two orthogonal lines in the original scene back to 90°. As you learned in Lecture 5, you can obtain the expression for $cos(\theta)$ in the form of Dual Degenerate Conic $\boldsymbol{C}^*_\infty$ as follows:

$$cos(\theta) = \frac{\boldsymbol{l}^T \boldsymbol{C}^*_\infty \boldsymbol{m}}{\sqrt{(\boldsymbol{l}^T \boldsymbol{C}^*_\infty \boldsymbol{l})(\boldsymbol{m}^T \boldsymbol{C}^*_\infty \boldsymbol{m})}} \tag{1}$$

where $\boldsymbol{C}^*_\infty = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Substituting $\boldsymbol{l} = \boldsymbol{H}^T \boldsymbol{l}'$ and $\boldsymbol{m} = \boldsymbol{H}^T \boldsymbol{m}'$ in the numerator and setting it to zero (i.e. $\theta = 90°$), we get the following constraint for estimating $\boldsymbol{H}$:

$$cos(\theta) = \boldsymbol{l}'^T \boldsymbol{H} \boldsymbol{C}^*_\infty \boldsymbol{H}^T \boldsymbol{m}' = 0 \tag{2}$$

where $\boldsymbol{H}$ is an affine transformation matrix of the form shown in Eq. 3.

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} = \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{3}$$

For affine distortion removal, you need to identify orthogonal line pairs, i.e., angle-to-angle correspondence between orthogonal lines in the original scenes and in the image planes. You will need **at least two** such orthogonal line pairs in order to solve the unknowns in the affine transformation matrix.

1.3 The One-Step Method

The one-step method removes both the projective and affine distortions in one step. Let $\boldsymbol{C}_{\infty}^{'*}$ be a projection of the dual degenerate conic $\boldsymbol{C}_{\infty}^{*}$, then the one-step method eliminates both distortions using the homography that maps $\boldsymbol{C}_{\infty}^{'*}$ back to $\boldsymbol{C}_{\infty}^{*}$.

Recall that the projection of the dual degenerate conic can be written as follows:

$$\boldsymbol{C}_{\infty}^{'*} = \boldsymbol{H}\boldsymbol{C}_{\infty}^{*}\boldsymbol{H}^{T} \tag{4}$$

Substituting Eq. 4 into Eq. 2 we get Eq. 5.

$$cos(\theta) = \boldsymbol{l}^{'T}\boldsymbol{C}_{\infty}^{'*}\boldsymbol{m}' = 0 \tag{5}$$

Now, let

$$\boldsymbol{C}^{'*} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}$$

then we can estimate the unknowns $a, b, c, d,$ and e by solving Eq. 5 with multiple pairs of $(\boldsymbol{l}', \boldsymbol{m}')$.

To do so, you need to identify the lines $(\boldsymbol{l}', \boldsymbol{m}')$ in an image such that their corresponding lines $(\boldsymbol{l}, \boldsymbol{m})$ in world coordinates are orthogonal. You need to identify **at least five** such pairs. Note that since we are working with HC where only ratios matter, you can fix $f = 1$.

After $\boldsymbol{C}^{'*}$ has been fully estimated, your task is now to derive the homography matrix $\boldsymbol{H}$ that satisfies Eq. 6.

$$\begin{aligned} \boldsymbol{C}^{'*} &= \boldsymbol{H}\boldsymbol{C}_{\infty}^{*}\boldsymbol{H}^{T} \\ &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{v}^{T} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{T} & \boldsymbol{v} \\ \boldsymbol{0}^{T} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}\boldsymbol{A}^{T} & \boldsymbol{A}\boldsymbol{v} \\ \boldsymbol{v}^{T}\boldsymbol{A}^{T} & \boldsymbol{v}^{T}\boldsymbol{v} \end{bmatrix} \end{aligned} \tag{6}$$

Subsequently, we can calculate $\boldsymbol{A}$ and $\boldsymbol{v}$ using Eq. 7. (read Lecture-5 for more details)

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^{T} &= \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix} \\ \boldsymbol{A}\boldsymbol{v} &= \begin{bmatrix} d/2 \\ e/2 \end{bmatrix} \end{aligned} \tag{7}$$

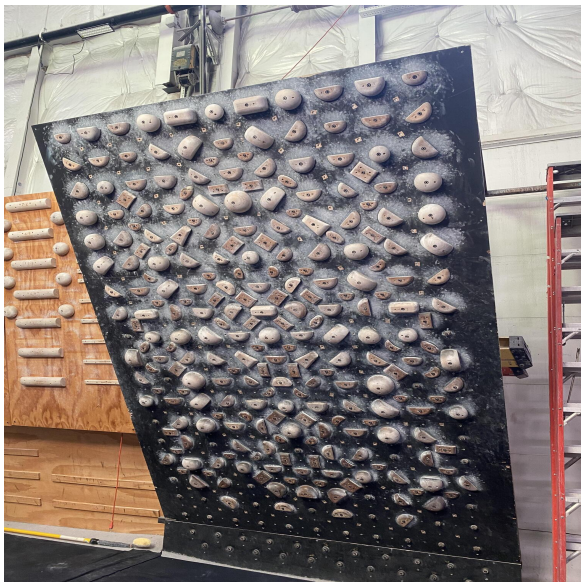
2 Programming Task

2.1 Task-1

Download the input images, shown in Fig. 1 and their world coordinates (height and width measurements of some planar object in the scenes), separately.

1. Show results after distortion corrections using **point-to-point correspondences**, this is a trivial extension of Homework 2. With the given height and width, your points in the undistorted image should be (0, 0), (0, width), (height, 0), and (height, width). After you have found the correspondences you simply need to apply the homography to the input image to remove distortion.
2. Show results after distortion correction using the **two-step** and **one-step** approaches.
3. Outline your observations on the results obtained using the above three methods.

(a) Image-1: Corridor – Use the metallic frames for drawing lines



(b) Image-2: Tension Board – Use the bolt holes for drawing lines

Figure 1: Input images for Task-1

Tension Board

The Tension Board is a specialized training tool widely used by climbers to enhance their skills. As shown in Fig. 1(b), the board is tilted, increasing the difficulty of climbing and introducing projective distortion in images. The board is equipped with various wooden holds, each securely fastened with bolts. By zooming

in Fig. 1(b), the bolt holes become visible and you can see that they are arranged in a precise grid pattern. This grid can be leveraged to draw orthogonal and parallel lines, aiding in the correction of any projective distortions in the captured images.

A mobile app accompanies the Tension Board, allowing users to select and set different climbing routes. When a route is chosen, the corresponding LEDs beneath the holds illuminate, indicating which holds are permitted for use. This feature adds a layer of structure and challenge to training sessions, making it an effective tool for focused practice. For a demonstration of how the board is used, refer to this video on YouTube titled "Board Lords Battle V15 on the Tension Board 2".

Other popular climbing boards, such as the Kilter Board and Moon Board, offer similar training functionalities with slight variations in design, catering to the diverse preferences of climbers. My personal favorite is the Kilter Board, but I did not have a picture of that for the homework.

2.2 Task-2

Repeat the steps, outlined in Task 1, on at least two of your own images. Usually, images with repetitive patterns on planar surfaces such as facades, walls with portraits, etc have many parallel or orthogonal line features and are easier to work with. Make sure to use images with significant projective and affine distortions. You can use approximate estimations of world coordinates, however state your assumptions clearly.

3 Submission Instructions

Include a typed report explaining how did you solve the given programming tasks.

1. Turn in a zipped file, it should include (a) a typed PDF report with source code files and results, (b) source code files (**only .py files are accepted — convert your .ipynb files to .py**), (c) Rename your .zip file as hw3_<First Name><Last Name>.zip and follow the same file naming convention for your PDF report too. There should be two items in your submission - (1) Homework PDF (2) ZIP file containing source code and text files (if any) for image coordinates.
2. Your PDF must include a description of the following:
   - The logic that you used to solve the given tasks.
   - The steps that you used for each of the tasks with relevant equations.
   - The input and output images for each task. Clearly show the plotted parallel / orthogonal lines that you chose in the input images.
   - At least some use vectorized numpy operations, if not fully optimized, is expected in your Python code.
   - Your observations on the output quality and performance of each approach.

- Your source code. Make sure that your source code files are adequately commented and cleaned up.

3. In order to avoid large file size of your submission, include JPEG images in your report for showing your results and your input images for Task2.

4. The sample solutions from previous years are for reference only, it's important not to get too biased by those solutions. **Your code and final report must be your own work**.

References

[1] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," 2003. 1, 2