Homework 07: Hough Transform; Optical Flow

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

Due: 2025-10-22, 11:59pm, on Canvas

General Instructions:

- You should solve the homework and submit your report **individually**. Identical submissions will receive a grade of zero.
- Getting help from others or checking your answers with other students (not the TAs) is okay and encouraged.
- Ask any questions on **Ed Discussion** (instead of emailing).
- **Before** the homework due date, TAs are strictly prohibited from **pre-grading** your homework. Do not expect the TAs to help you verify if your answers are correct or give you the problem solution.
- **After** the homework due date, if you do not know how to solve a problem, reach out to the TAs. They will walk you through the solution and help you understand it. Note that homework solutions will **not** be posted because some problems will be used in next year's class.
- **Exams** may contain questions related to homework, so make sure you learn how to solve the homework problems correctly.
- The deliverables are outlined for each problem, and you should carefully **follow the instructions**. Failing to follow instructions will result in **points being subtracted**.
- You will submit a **single PDF** file to Canvas as your homework report. The PDF must contain your **answers** and any requested **outputs** (e.g., printouts, snapshots of code, or GUIs). If requested, follow the instructions specified by the problem to provide your **code** (e.g., in a compressed .zip or .tar file) in addition to the PDF file.
- **Grading:** Each homework in this class will contribute **5pts** to your final grade (there will be 12 homework assignments, each 5pts, leading to 60pts for all assignments). A detailed grading **rubric** will be posted on **Canvas** after the homework due date. Any bonus points will be added to your overall course bonus points, which will be added to your final grade.
- **Late submission:** Late or missed submission will not be accepted and will receive a grade a zero. Any excused absence must be documented and disclosed to the instructor (extensions will be granted on a case-by-case basis). Three or more missed homework lead to an INC grade.

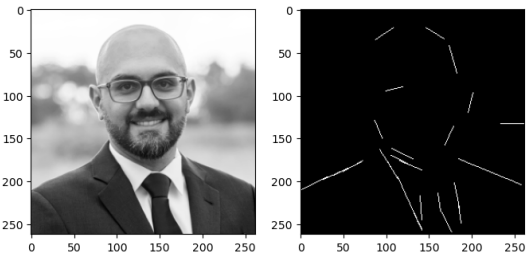
Homework 07: Hough Transform; Optical Flow

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

Due: 2025-10-22, 11:59pm, on Canvas

EXERCISE 1 (1pts) – Extract lines by using Hough transform on your ***headshot image***. Adjust algorithm parameters so that a ***minimum of 5 lines*** are detected. Display/overlap the detected lines on the image, e.g.,

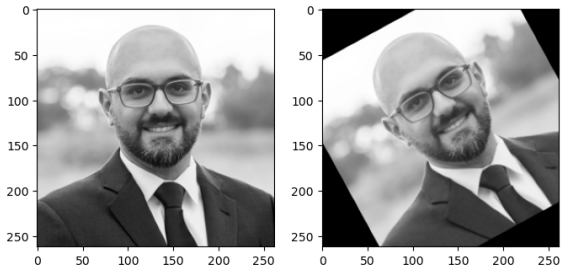


Deliverables:

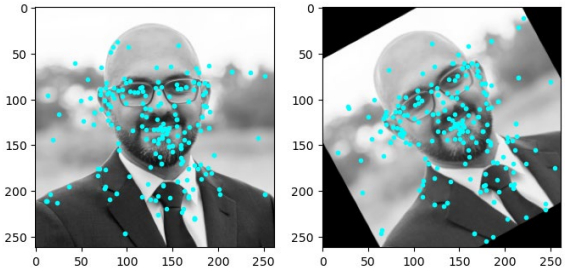
- Snapshot of entire code
- Input image
- Image of detected lines

EXERCISE 2 (2pts) – Create a transformed headshot image by rotating your image ***0.5 radians*** (or ***30 degrees***). Translate the image as needed to center it, e.g.,

```
tform = transform.AffineTransform(scale=(1,1), rotation=0.5, translation=(70, -50))
img2 = transform.warp(img, tform)
```



Extract SIFT keypoints and descriptors for each image. Adjust SIFT parameters to extract a ***minimum*** of ***10 keypoints***. Overlay and display the keypoints on the images, e.g.,



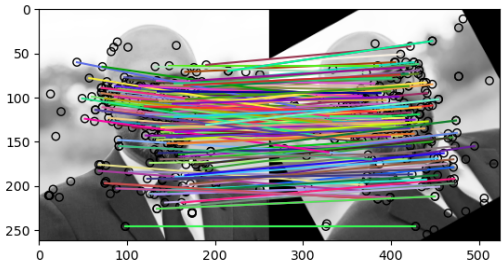
Extract SIFT descriptors and use them to match the keypoints across both images. You must use ***Low's ratio of 0.7*** to remove outlier matches and use ***bi-directional matching*** so that matches are across both images. Display matching across both images.

Homework 07: Hough Transform; Optical Flow

Kaveh Fathian, Email: fathian@ariarobotics.com

Handout: 2025-10-13

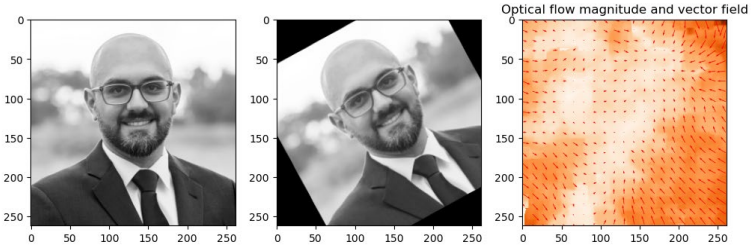
Due: 2025-10-22, 11:59pm, on Canvas



Deliverables:

- Snapshot of entire code
- Input image and its transformed version
- SIFT keypoints overlayed on images
- Matched SIFT keypoint image

EXERCISE 3 (2pts) – Compute the optical flow using Lucas-Kanade algorithm between your image and its transformed version in Exercise 2. Display the optical flow as an image (using appropriate color map), or display the optical flow using quiver vectors and use a color map to display its magnitude, e.g.,



Is the optical flow computed correctly for all pixels in the image? Are there pixels for which the optical flow is wrong? If yes, explain why.

Deliverables:

- Snapshot of entire code
- Optical flow image
- Answer to questions and explanation of optical flow correctness