Homework 2 Written Questions

Template Instructions

This document is a template with specific answer regions and a fixed number of pages. Given large class sizes and limited TA time, the template helps the course staff to grade efficiently and still focus on the content of your submissions. Please help us in this task:

- Make this document anonymous.

- Questions are in the orange boxes. Provide answers in the green boxes.

- Use the footer to check for correct page alignment.

- **Do NOT remove the answer box.**

- **Do NOT change the size of the answer box.**

- **Extra pages are not permitted unless otherwise specified.**

- **Template edits or page misalignment will lead to a 10 point deduction.**

Gradescope Submission

- Compile this document to a PDF and submit it to Gradescope.

- Pages will be automatically assigned to the right questions on Gradescope.

This Homework

- 6 questions **[10 + 9 + 9 + 7 + 3 + 5 = 43 points]**.

- Include code, images, and equations where appropriate.

Declaration of Generative AI Use

Reminder of Course Policy

- The use of GenAI tools (e.g., ChatGPT, Grammarly, Bard) for completing any part of this course is discouraged.

- Using these tools is not needed to be successful in the class and could be detrimental to your learning experience.

- If you use them, you must cite the tool you used and explain how you used it.

- If you do not cite the tool, it is an academic code violation.

- We will be using special tools to detect cases of unattributed GenAI use.

Student Declaration

Have you used generative AI tools to complete this assignment:

YES ☐ NO ■

If you answered YES above, describe what tools you used and what parts of the assignment you used them for below:

Example: I used ChatGPT to debug my convolution implementation

Q1:   [10 points]

Let's look again at the webcam Fourier decomposition demo that we saw in class. The Fourier transform lets us operate on images with respect to their frequency content.

Navigate to the `questions` folder. Then, within your virtual environment, run the following command:

```
$ python liveFFT2.py
```

*Note: Your computer must have a webcam. If it does not, find the **use_webcam** variable and set it to false. If you're running an exotic setup, like from within WSL, then your webcam might not be accessible.*

The python file contains five parts for you to explore the effect of manipulating the Fourier decomposition representation upon the reconstructed image. In the demo, top left is the input image, bottom left is the basis frequency amplitude image, bottom right is the corresponding phase image, and top right is the reconstruction of the input image via the Fourier representation.

- Part 0: Scanning the basis and observing the output image.

- Part 1: Reconstructions from different numbers of basis frequencies.

- Part 2: Replacing amplitude and phase with that from a different image.

- Part 3: Replacing amplitude and phase with that from a noise image.

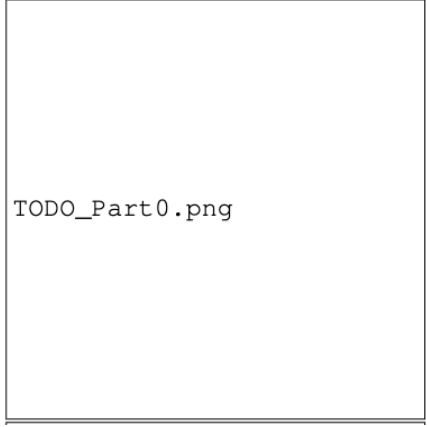- Part 4: Manipulating the amplitude and phase images.

Uncomment the different parts and explore! When uncommenting, only uncomment one part at a time.

You **will** have to answer short responses and screenshots in the following parts so you might want to answer them as you go!

Note: Please keep grading anonymous. Use your cat as a model, show your favourite vector calculus book, etc. Extra credit for creative effort.

In the following questions, please include a screenshot for the given part as well as answer a short response. **[1-2 sentences]**.
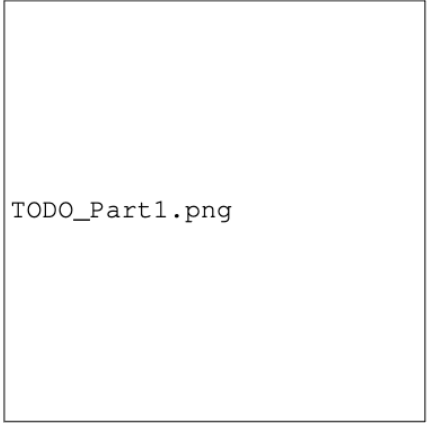
(a) **[2 points]** In **Part 0**, the reconstructed image (top right) looks like bars of black and white lines rather than the image to the left of it. Why is this? Hint: Look what we do after zero-ing out the amplitude and phase.

TODO_Part0.png

TODO: Your answer for (a) here

(b) **[2 points]** In **Part 1**, what is the relation between how many bases are applied and the reconstructed image (top right image)?

TODO_Part1.png

TODO: Your answer for (b) here

Only Q1 (a) - (b) should be on this page   4 / 17

(c) **[2 points]** In **Part 2** and **Part 3**, after replacing the phase/amplitude with the phase/amplitude of the image of Jack or the noisy image, which do you think contributes more to the reconstruction?

TODO_Part2.png   TODO_Part3.png

TODO: Your answer for (c) here

(d) **[2 points]** After **Part 4**, describe the effect of one modification you saw in the demo after playing around with the phase or amplitude in relation to the reconstructed image.

TODO_Part4.png

TODO: Your answer for (d) here

(e) **[2 points]** In the demo you saw how the Fourier Transform deconstructs an image into amplitude and phase, then subsequently transforms it back into a reconstructed image. This process has a fairly nifty use-case in relation to convolution that we went over in lecture. In your own words, what is The Convolution Theorem? More specifically, how does this theorem along with the Inverse Fast Fourier Transform provide a beneficial use case in relation to convolution time complexity? [2-3 sentences]

TODO: Your answer for (e) here

Q2: **[9 points]** Often in computer vision, we wish to find points in one image that match to the same world point in another image—so-called correspondence finding.

> In the following three pairs of images, please use the included python script `plot_corners.py` to find corners using the Harris corner detection algorithm. For each pair, discuss any differences in the returned corners, the properties of the images that create these differences, and any underlying real world phenomena that may have been the primary cause of these differences. **[3–5 sentences]**

(a) **[3 points]** RISHLibrary: RISHLibrary1.jpg and RISHLibrary2.jpg

TODO_RISHLibrary1.png   TODO_RISHLibrary2.png

TODO: Your answer for (a) here

(b) **[3 points]** Chase: Chase1.jpg and Chase2.jpg

TODO_Chase1.png   TODO_Chase2.png

TODO: Your answer for (b) here

(c) **[3 points]** LaddObservatory: LaddObservatory1.jpg and LaddObservatory2.jpg

TODO_Ladd1.png   TODO_Ladd2.png

TODO: Your answer for (c) here

Q3: **[9 points]** One use of local feature description and matching is fingerprint recognition—please watch the video and note the similarity to our task in this homework.

Fingerprint recognition uses biometric data—measurements of human biological features that are unique to an individual—to make it convenient to unlock doors or devices quickly and without needing to remember a password. However, given its uniqueness, biometric data may be seen as a greater privacy encroachment upon a person than a password. Further, given the trust that is derived from the uniqueness of biometric data, it may also pose a greater risk of misuse if the data is not secure because the data cannot be changed.

(a) **[3 points]**

> Do you use biometric recognition systems? List them. [If not, list some that people around you use.] For one of the systems you use, where is the reference data stored (such as your stored fingerprint), and where and how does the authentication process happen (at a high level)? Try to find the answer online; this article may also help. **[4–6 sentences]**

TODO: Your answer for (a) here

(b) **[3 points]**

> How might someone use computer vision to steal or spoof your biometric data to gain access? *This could be across reconstruction, recognition, or (re)organization.* **[3–5 sentences]**

TODO: Your answer for (b) here

(c) **[3 points]**

> Biometric recognition systems may not affect all people equally. For a biometric identification system, define a group of people and describe how they might be affected disproportionately. **[3–5 sentences]**

TODO: Your answer for (c) here

Q4: **[7 points]** Brown University decides to entirely replace passwords with biometric data to authenticate student identity on its computer systems. Given how accurate your feature matching homework 2 code is, Brown asks you to develop the authentication system as your CSCI 1430 final project. Lucky you.

In preparation, you read a previous case about a biometric data breach.

(a) **[3 points]**

> How were BioStar 2 storing their fingerprint data? Knowing the computer vision algorithms involved in feature matching, what different processing, features, or storage might you consider instead to decrease the risk of a biometric data breach? **[4–6 sentences]**

TODO: Your answer for (a) here

(b) **[4 points]** Even though fingerprints are thought to be unique, we are bound by the accuracy of computer vision systems to detect and recognize that uniqueness. This may be a challenge for Brown's 10,000 students, let alone a national-scale database such as the FBI's Next Generation Identification System that houses over 100 million fingerprints; its Advanced Fingerprint Identification Technology is claimed to be 99.6% accurate.

Even seemingly high accuracies can create many inaccurate matches with large databases, potentially causing inaccurate judgements in criminal cases.

> Fingerprint images are quite different from the images of natural 'human-scale' scenes we've been using throughout the course. Is correspondence finding for fingerprint images an easier problem than for natural images? Why or why not? Think about the variation within each domain of images, the accuracy required, the consequences of a match (or lack thereof), the accuracy required, and the scope of the application. **[6–8 sentences]**
> *Refer to the video at the beginning of Q3 and the first 15 slides of this deck for example fingerprint images and additional information.*

TODO: Your answer for (b) here

Q5: **[3 points]** As we saw in Q2, the Harris Corner Detector can find feature points for image matching.

To detect these features, the algorithm approximates the strength of any change in the auto-correlation function $E(u, v)$ over shifts u, v. We look at the second derivative term in the Taylor series expansion of E to consider the local shape of $E(u, v)$ via the *structure tensor* M. Then, we analyze M to determine if that pixel is a corner or not.

Note: We expect this conceptual question to require careful study of the lecture slides.

> How do the eigenvalues of the M structure tensor vary with local image brightness, and how might we interpret the eigenvalues geometrically as an ellipse)?
> **6–8 sentences**

TODO: Your answer for Q5 here

Q6: **[5 points]** The SIFT algorithm can extract feature descriptors around feature points. Given a feature point location, the SIFT algorithm converts a 16×16 window around the feature point into a 128×1 feature descriptor via a radial histogram of the gradient magnitudes within the window.

> Write pseudocode *but with correct matrix/array indices* for descriptor extraction. *Notes:* Do this for just one interest point at one scale; ignore the overall interest point orientation; ignore the Gaussian weighting; ignore all normalization post-processing; ignore image boundaries; ignore sub-pixel interpolation and just pick an arbitrary center within the 16×16 for your feature descriptor.

```
# You can assume access to the image, x and y gradients,
# and their magnitudes/orientations.
image = imread("rara.jpg")
grad_x = filter(image, "sobelX")
grad_y = filter(image, "sobelY")
grad_mag = sqrt(grad_x .^2 + grad_y.^2)
grad_ori = atan2(grad_y, grad_x)

# Takes in a interest point x,y location and returns
# a feature descriptor
def SIFTdescriptor(x, y)
    descriptor = zeros(128,1)

    # TODO: Populate descriptor with the right gradient
    # magnitudes dependent on the gradient orientations
```

```
    return descriptor
```

Feedback? (Optional)

Please help us make the course better. If you have any feedback for this assignment, we'd love to hear it!