Homework 2 Written Questions

CSCI 1430

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.

Only instructions should be on this page

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 six parts for you better understand the Fourier transform and how manipulating the Fourier decomposition representation affects 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: Decomposition and reconstruction of camera feed.
- Part 1: Scanning the basis and observing the output image.
- Part 2: Reconstructions from different numbers of basis frequencies.
- Part 3: Replacing amplitude and phase with that from a different image.
- Part 4: Replacing amplitude and phase with that from a noise image.
- Part 5: Manipulating the amplitude and phase images.

Within `studentFuncs()`, uncomment the different parts and explore!

You will answer short responses and capture screenshots of the following parts, so you might want to answer them as you go!

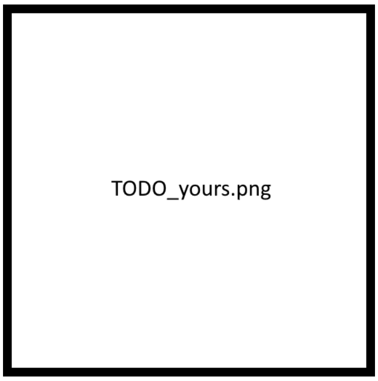
Note: Please keep grading anonymous. Use your cat as a model, show your favourite vector calculus book, etc. Extra credit for creative effort.

Only Q1 should be on this page

In the following subquestions, please include a screenshot for the given part as well as answer a short response. **[1-2 sentences]**.

Q1.1 — [2 points]

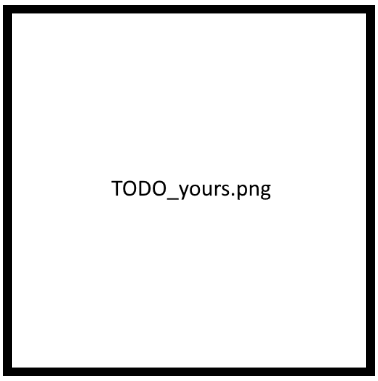
In **Part 1**, the reconstructed image (top right) looks like bars of black and white lines. Why is this? Hint: use 'm' and 'o' on your keyboard to animate the direction and distance from center of the dot in the bottom left.



TODO: Your answer for (a) here

Q1.2 — [2 points]

In **Part 2**, what is the relation between how many bases are applied and the reconstructed image?

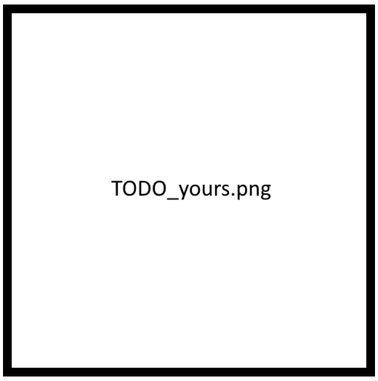


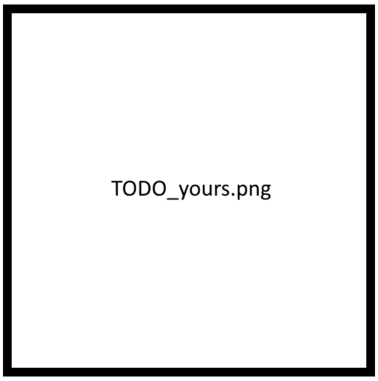
TODO: Your answer here

Only Q1.1, Q1.2 should be on this page

Q1.3 — [2 points]

In **Part 3**, we replace our input image phase with that of an image of Merlin the cat. In **Part 4** we replace the amplitude with noise. Which of phase or amplitude contributes more to your perception of the reconstructed image and why?

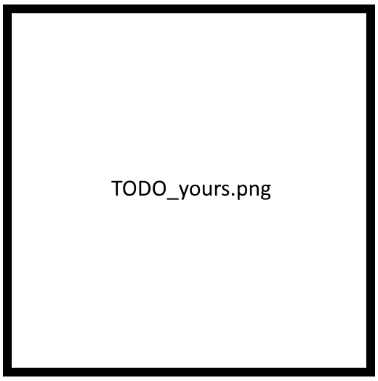




TODO: Your answer here

Q1.4 — [2 points]

After **Part 5**, describe the effect of one modification you saw in the demo after playing around with the phase or amplitude.



TODO: Your answer here.

Only Q1.3, Q1.4 should be on this page

Q1.5 — [2 points]

In the demo, we use the Fourier Transform to deconstruct an image into amplitude and phase, then reconstruct those back into an image. This process has a connection to convolution as we discussed in lecture. In your own words, what is The Convolution Theorem and what benefits can it provide? **[2-3 sentences]**

TODO: Your answer here

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.

Q2 — [9 points]

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. Run the script with `python plot_corners.py {example}` where `{example}` is one of {`RISHLibrary`, `Chase`, `LaddObservatory`}. 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]**

RISHLibrary: [3 points] RISHLibrary1.jpg and RISHLibrary2.jpg

TODO_RISHLibrary1.png

TODO: Your answer here

Only Q2.1 should be on this page

Chase: [3 points] Chase1.jpg and Chase2.jpg

TODO_Chase1.png

TODO: Your answer here

LaddObservatory: [3 points] LaddObservatory1.jpg and LaddObservatory2.jpg

TODO_Ladd1.png

TODO: Your answer here

Only Q2.3 should be on this page

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 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.

Q3.1 — [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), where does the authentication happen, and how does the authentication happen (at a high level)? If you don't know, then try to find the answer online. **[4–6 sentences]**

TODO your answer here.

Q3.2 — [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 here.

Q3.3 — [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 here.

Only Q3.2, Q3.3 should be on this page

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.

Q4.1 — [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 here.

Q4.2 — [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 (since 2014); 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 for more detail please refer to the first 15 slides of this deck—it has example fingerprints and additional information.

TODO: Your answer here.

Only Q4.2 should be on this page

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. M is also called the second moment matrix.

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 here

Only Q5 should be on this page

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
```

Only Q6 should be on this page

Feedback? (Optional)

We appreciate your feedback on how to improve the course. You can provide anonymous feedback through this form which can be accessed using your Brown account (your identity will not be collected). If you have urgent non-anonymous comments/questions, please email the instructor.