Homework 2 Written Questions

Document Instructions

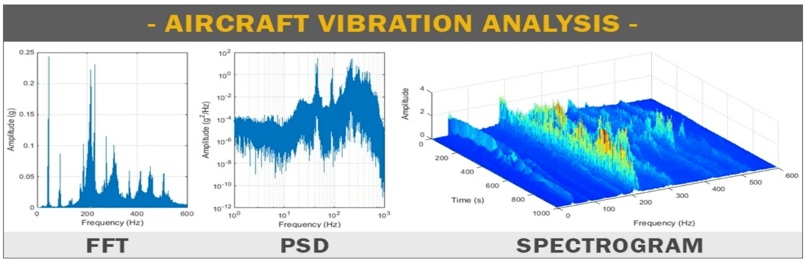
- 6 questions **[3 + 11 + 9 + 6 + 3 + 3 = 35 points + 2 bonus points]**.
- Fill all your answers within the answer boxes, and **please do NOT remove the answer box outlines**.
- Questions are highlighted in the **orange boxes**, bonus questions are highlighted in **blue boxes**, answers should be recorded in the **green boxes**.
- Include code, images, and equations where appropriate.
- To identify all places where your responses are expected, search for 'TODO'.
- The answer box sizes have been set by the staff beforehand and **your responses should not exceed the green borders**. Any overfull text may be truncated, so make sure your responses fit. **Extra pages are not permitted unless otherwise specified.**
- Make sure your submission has the right number of pages to validate page alignment sanity (check the footer).
- Please make this document anonymous.

Gradescope Instructions

- When you are finished, compile this document to a PDF and submit it directly to Gradescope.
- The pages will be automatically assigned to the right questions on Gradescope *assuming you do not add any unnecessary pages*. **Inconsistently assigned pages will lead to a deduction of 2 points per misaligned page (capped at a maximum 6 point deduction).**

Q1: **[3 points]** Let's look again at the webcam Fourier decomposition demo which we showed in class.

The Fourier transform (or fast Fourier transform more specifically) is often regarded as one of the most important algorithms ever developed. The internet, WiFi, phones, computers, routers, satellites, and almost everything that has a computer inside uses these algorithms in one way or another. Below is a depiction of FFT being used for aircraft vibration analysis.



Navigate to the `questions` folder. Then, within your cs1430 virtual environment, run the following command:

```
$ python liveFFT2.py
```

Warning: If you're running the code from within WSL, this will likely not work since you need access to your laptop's camera / webcam.
This file contains five parts for you to explore and see the amplitude image, the phase image, and the effect of the reconstructed image.

- 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 the camera feed decomposition! Please include five screenshots documenting your experimentation with each of the five sections. We'll be grading for completion, not correctness.
Note: For anonymous grading, try not to put yourself in the camera frame. Show your favourite vector calculus book, wear a mask, use your cat, etc. Extra credit for creative effort.

TODO_Part0.png

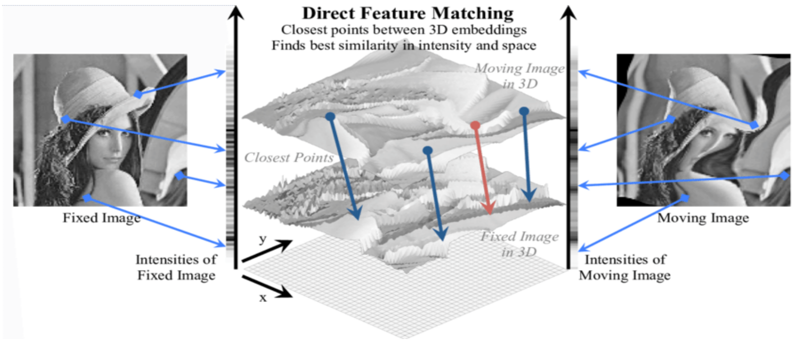
TODO_Part1.png

TODO_Part2.png

TODO_Part3.png

TODO_Part4.png

Q2: [11 points] Imagine we wished to find points in one image which matched to the same world point in another image - so-called feature point correspondence matching. We are tasked with designing an image feature point algorithm which could match world points in the following three pairs of images.



(a) **[9 points]**

Please use the included python script `plot_corners.py` to find corners using Harris corner detection for each pair of images below.
For each pair, discuss the differences in the returned corners (if any), the aspects 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]**

(i) **[1 + 2 points]** RISHLibrary: RISHLibrary1.jpg and RISHLibrary2.jpg

TODO_RISHLibrary1.png	TODO_RISHLibrary2.png

TODO: Your answer for a(i) here

(ii) **[1 + 2 points]** Chase: Chase1.jpg and Chase2.jpg

TODO_Chase1.png

TODO_Chase2.png

TODO: Your answer for a(ii) here

(iii) **[1 + 2 points]** LaddObservatory: LaddObservatory1.jpg and LaddObservatory2.jpg

TODO: Your answer for a(iii) here

(b) **[2 points]**

In the coding portion of the homework, you'll investigate strategies to filter only the matches you are most confident in. In the context of a real-world example, explain what problems could be caused if someone took feature matching algorithm results to be 100% accurate? **[3-4 sentences]**

TODO: Your answer for (b) here

Q3: [9 points] One use of local feature extraction 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) **[2 + 1 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 authentication system, define a group of people and describe how they might be affected disproportionately. **[3-5 sentences]**

TODO: Your answer for (c) here

Q4: [6 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) **[1 + 2 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

Only A4 (a) should be on this page

(b) **[2 + 1 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 inaccurage judgements in criminal cases.

What are the consequences of feature matching for biometric authentication when compared to feature matching across photographs? What responsibilities do the creators of these technologies have, if any, to ensure accuracy or reliability? **[5-7 sentences]**
Refer to the video at the beginning of Q3 and this slide deck for example images and additional information.

TODO: Your answer for (b) here

Q5: **[3 points]** Consider the pictures below of the same mountain.


If you were tasked with the problem of stitching them together to form a panoramic view, what would be the first thing you would think of doing? You would probably find points in one image which appear in the other.

The Harris Corner Detector is commonly used in computer vision algorithms to find interest points from which to extract stable features for image matching.

To detect these features, we attempt to determine the strength of any change in the Energy (intensity) function $E(u, v)$ on the image. Since we can approximate corner points to be represented by quadratic-esque energy level surfaces, we can use the Taylor expansion and quantify the change in $E(u, v)$ within a certain radius in the image.

You might remember from your Calculus class that the second partial derivative test helps us determine if a point is a local min, local max or saddle point. Similarly, we focus on the second derivative term in the Taylor expansion to determine the shape of $E(u, v)$ and determine if that particular point is a corner or not.

Now, we can represent the Taylor expansion using matrices, and so the second derivative can be expressed as a matrix. Specifically, this matrix, M, is called the second moment matrix.

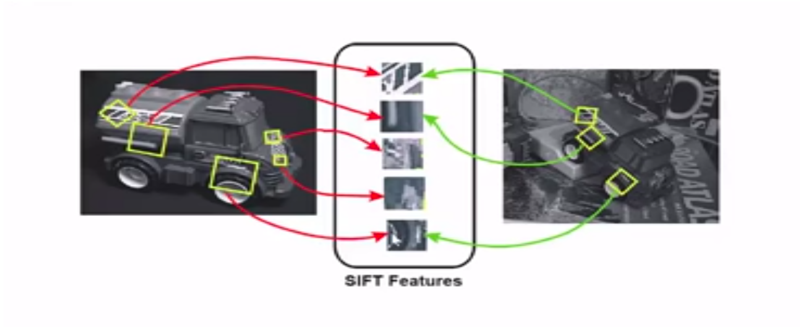
Note: For a more comprehensive background, review the lecture slides.

How do the eigenvalues of the 'M' second moment matrix vary with local image brightness, and how might we interpret the eigenvalues geometrically (think 'shape')? **6-8 sentences**

TODO: Your answer for Q5 here

Q6: **[3 points]** Once you've detected the feature points, the next step is to extract their descriptions, i.e., their general properties. These generated feature descriptors will be used in the final step when matching feature points via their feature descriptors.

We use the SIFT algorithm to extract feature descriptors around feature points. Given a feature point location, the SIFT algorithm converts a 16×16 patch around the feature point into a 128×1 feature descriptor of the gradient magnitudes and orientations therein.



Write pseudocode *with matrix/array indices* for these steps.
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











    ################################################
```

```
    # YOU MAY USE THIS ADDITIONAL PAGE

    # WARNING: IF YOU DON'T END UP USING THIS PAGE
    # KEEP THESE COMMENTS TO MAINTAIN PAGE ALIGNMENT
    ################################################

    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!