Homework 2 Questions

Instructions

- 2 socially-responsible computing questions, which will be expanded on in discussion sections.
- 1 exercise graded for completion only.
- 3 graded technical questions.
- Write code where appropriate.
- Feel free to include images or equations.
- Please make this document anonymous.
- This assignment is **fixed length**, and the pages have been assigned for you in Gradescope. As a result, **please do NOT add any new pages**. We will provide ample room for you to answer the questions. If you *really* wish for more space, please add a page *at the end of the document*.
- **We do NOT expect you to fill up each page with your answer.** Some answers will only be a few sentences long, and that is okay.

Q1: 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. Further, given the trust that is derived from its uniqueness, biometric data may also pose a greater risk of misuse if the data is not secure because the data cannot be changed.

a) Do you use biometric recognition systems? List them. [If not, suppose that you do.] For each, where is the reference data stored (such as your stored fingerprint) and where does the authentication process happen?

b) How might someone use computer vision to steal or spoof your biometric data to gain access?

c) Biometric recognition systems may not affect all people equally. For a biometric authentication system, define and describe how a specific group of people might be affected disproportionately.

A1: Your answer here.

- national or global-scale mismatches

Q2: 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.

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

1) How were BioStar 2 storing their fingerprint data? Knowing the algorithms involved in feature matching, what could you do instead from a computer vision perspective to decrease the risk of storing biometric data?

2) Even though fingerprints are thought to be unique, we are bound by the accuracy of computer vision systems to detect and recognize that uniqueness. Brown has approximately 10,000 students.

A2: Your answer here.

Exercise

E1: Let's look again at the webcam Fourier decomposition demo which James showed in class. Let's run it in our CSCI 1430 virtual environment *preferably on a computer with a webcam*, from within the 'questions' directory.

```
$ python liveFFT2.py
```

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 the results of your experimentation, e.g., two-to-three screenshots of what you discover. 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.

Exercise Results

Please include images of your results from the exercise here, e.g., two-to-three screenshots of your findings.

Extra space

Q3: 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.

RISHLibrary: One Two — *Chase:* One Two — *LaddObservatory:* One Two

Please use the included python script `plot_corners.py` to find corners using Harris corner detection for each image above.

For each pair, discuss the differences in the returned corners (if any), and what aspects of the images may have caused these differences. Then discuss what real world challenges exist in matching features between images.

A3 (a): Answer for RISHLibrary pair

A3 (b): Answer for Chase pair

A3 (c): Answer for LaddObservatory pair

Q4: The Harris Corner Detector is commonly used in computer vision algorithms to find interest points from which to extract stable features for image matching.

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’)?

A4: Your answer here.

Q5: Given a interest point location, the SIFT algorithm converts a 16×16 patch around the interest 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. Please just explain in pseudocode how to go from the 16×16 patch to the 128×1 vector. You are free to simplify the gradient computation.

A5: Your answer here.

```
# 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)

    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!