MIT CSAIL

6.869 Advances in Computer Vision

Fall 2018

Problem Set 4

Posted: Thursday, October 4, 2018 **Due:** Thursday October 11, 2018

Submission Instructions: Please submit **three separate files:** 1) a report named ⟨your_kerberos⟩.pdf, including your responses to all required questions with images and/or plots showing your results, 2) a video file named ⟨your_kerberos⟩.avi containing your output for Problem 3d, and 3) a file named ⟨your_kerberos⟩.zip, containing relevant source code. **Submissions that do not adhere to these instructions are subject to an additional penalty.**

Late Submission Policy: We do not accept late submissions. The submission deadline has a 50-minute soft cut-off; after midnight Thursday, submissions are penalized 2% per minute late.

Collaborators: You are free to discuss problems with other students but all writing must be done individually. Please list all collaborators at the top of your report.

Problem 1 *Hybrid images*

In this problem you will create hybrid images as described in [1]. You may find the tools [2] useful.

Take two images, A and B, that you'll want to have blend from one to the other. Try to make the objects in the two images occupy more or less the same region. Construct a hybrid image from A (to be seen close-up) and B (to be seen far away) as follows:

$$\text{out} = \texttt{blur}(\text{B}) + (\text{A-}\texttt{blur}(\text{A}))$$

Where `blur` is a function that low-pass filters the image. You should write your own blur function. You can use a Gaussian filter or try other blur filters, such as the box filter. Which one works best? Try different sigmas for the Gaussian. How does the amount of blurring affect your perception of the results?

In your report, please specify the type of kernel you used and its parameters. Also, attach your final result.

Problem 2 *De-hybridizing*

Examine the image `einsteinandwho.jpg` included with the problem set. Using the method of your choice, remove the individual represented in the low spatial frequency range to create two images: one of Einstein, and one of the other person. Please intensity scale the images using the provided code to make them easier to see. Include both images and your code in the report. You may want to try different methods to achieve the best two images. For fun: can you guess who is in the low spatial frequency image?

Problem 3 *Motion Magnification*

In this problem we will investigate motion magnification in videos. Recall that position shifts in image space correspond to phase shifts in the frequency domain of the Fourier transform. This means that for two images, we can compare the Fourier transform of the two images to find the phase shift between the images. Amplifying the phase shift by a fixed factor in the Fourier transform frequency domain will amplify the position shift by the same factor in the image domain after we perform the inverse Fourier transform. We will use this idea to exaggerate the motions in videos.

(a) For a purely horizontal offset of an impulse signal, magnifying the phase shift will result in a magnified horizontal offset after the inverse transfom. Please fill in lines 8 and 11 in `magnifyChange.m`. You should find the phase shift between the two input images and magnify it by the specified `magnificationFactor`. When complete, the function `magnifyChange` should return an image showing what image 2 would look like with the magnified offset. Please run `part_a.m` and submit the generated plot.

(b) If there is motion in more than one direction between two images, we will see that naively magnifying the phase shift of the whole images will not work. In `part_b.m`, we have set up a vertical offset of an impulse signal as well as the horizontal one from part a. Please run `part_b.m` and submit the generated plot, then explain why the two offsets were not properly magnified.

(c) One strategy we can use if there are multiple motions between two images is to do a localized Fourier transform by independently magnifying the offsets on small windows of the images and aggregating the results across the windows. When we restrict our window of consideration, it is more likely for everything in the window to be moving the same way. We will use Gaussian filters to mask small windows of the image and perform magnification on each window independently. In `part_c.m`, please fill in the Gaussian filter in line 30 and the appropriately windowed input images in line 32. Since we are working with images, we will use the discrete Gaussian filter rather than the continuous one. Run `part_c.m` to confirm that the two motions were properly magnified and submit the generated plot.

(d) We are now ready to apply motion magnification to videos. We will use the same approach as in part c of magnifying Gaussian windowed regions of the video frames. Rather than directly finding the phase shifts between consecutive video frames, we will keep a moving average of the Fourier transform phases and compare each new frame's DFT phase with the current moving average of phase. The moving average is an IIR low-pass filter, averaging 0.5 times the previous average with 0.5 times the current phase. For simplicity, each of the

RGB channels are processed independently and identically. In `video_momag.m`, you will need to fill in the Gaussian filter in line 49, the DFT phase of the magnified window in line 69, and the DFT of the magnified window in line 72. Please run `video_momag.m` and submit the generated video. Note that the code may take some time to run - you can temporarily modify `sigma` to decrease the number of windowed regions to process.

References

[1] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. *ACM Transactions on Graphics (TOG)*, 2006.

[2] Eero Simoncelli. Matlab pyrtools. `http://www.cns.nyu.edu/~eero/software.php`.