October 17, 2020

PROJECT 1 REPORT

Eric Miao
CS-7390
Fall 2020
hongyum@smu.edu

1. ABSTRACT

This is a project that focus on using different OpenCV image filtering functions and use this technique to create hybrid images. The theory behind is based on a SIGGRAPH 2006 paper about hybrid images.

2. INTRODUCTION AND MOTIVATION

The motivation behind hybrid images is related to studies in human perception. In the early 1970s, sisual psychophysics research has shown that human observers are able to comprehend the meaning of a novel image within a short glance. Later, in 1990s, more research implies that in the early visual processing, human is performing a coarse-to-fine frequency analysis on the image. When the task required identifying a scene image quickly, human observers interpreted the low spatial frequency band before the high spatial frequency band.

Moving forward, there are more and more information demonstrates that the selection of frequency bands for fast image recognition is a flexible mechanism for human. In its simplest version, a hybrid image can be made by overlapping one coarse, blurry picture with one fine, detailed picture. By using hybrid image, we can provide a new paradigm that images interpretation can be modulated by playing with viewing distance or presentation time (Hutchison, 2012).

Besides, there are practical applications in the real world that is based on the concept of hybrid image. For example, private font uses the hybrid images to display text that is not visible for people standing at close distance. I also believe that it could be a great tool in machine learning, as a few scientists had developed methods that use hybrid image to bridge the gap between real and simulation environments for robotics (Bayraktar, 2018).

In this project, I created two interactive tools to blend images and see the visual results. User also has the option to save the selected results in an image file.

3. BACKGROUND AND CONTEXT

Images we see today is static, and every image is made up of different components at different spatial frequencies, which capture different aspects of that image. Low spatial frequencies depict global luminance variations in the image and broad contours. When viewed alone, the low spatial frequencies of an image are literally a blurry, out of focus version of that image, and the high spatial frequencies of an image resemble a highly detailed line drawing (Oliva, 2013).

By blending the high-frequency portion of one image with the low-frequency portion of another image, we get a hybrid image that leads to different interpretations at different distances. Similarly, we can convert static images to their frequency domains and do the similar merging process with their magnitude and phase. After converting the blended image back to the spatial domain, we can get another kind of hybrid image.

4. METHODS

A. Image A (Program A)

i. Solution overview: I breakdown my solution to four steps: read images, get kernel, process images, and display images.

1. *Read images: The user interface will list 11 images from our current directory. User has the option to choose which two images he or she would like to use in each mode. The program also comes with error handling. If user's input is invalid or the image is not found, the program will also notify the user.*

2. *Get Kernel: After reading in the images, the program will compute for kernel that will be used later in the convolution process. If user chooses Mode 1, user has the ability to slide between 0 to 100 to modify the kernel size. However, due to performance issue, user cannot use slider to modify the kernel size in Mode 2 even though Mode 1 and Mode 2 are doing the same thing. The program will receive the kernel filter using the getGaussianKernel function.*

3. *Process images: After reading in and receive the kernel, the kernel will start the convolution process:*
   High frequency image = Original Image – Low Frequency Image
   Low frequiency image = highPassFilter(Original Image)
   newImg(channel) = High frequency(channel) + Low frequiency(channel)

4. *Display images: The program will display the hybrid image with the slider control, the low frequency image, and the high frequency image. When sliding between different values, all images will change at the same time.*

ii. Sliders control: At first, I was planning on using two sliders. One for the high frequency image and one for the low frequency image. In my final implementation, I minimize my variables and use 1 slider control instead. The slider control is called blend that allows user to slide from value 0 to 100. When value is closer to 0, the hybrid image would look more like the low frequency image. Similarly, when the blend value is closer to 100, the hybrid image would look more like the high frequency image.

B. Image B (Program B)

i. Solution overview: I breakdown my solution to four steps: read images, convert images to frequency domain, process images, and display images.

1. *Read images: Like Program A (Mode 1 and Mode 2), the user interface will list 11 images from our current directory. User has the option to choose which two images he or she would like to use in each mode. The program also comes with error handling. If user's input is invalid or the image is not found, the program will also notify the user.*

2. *Convert images to frequency domain: I used discrete Fourier transform to get the complex image of each images. Then, I separate the magnitude and the phase from each complex image and store them in different mat objects.*

3. *Process images: I reconstructed a new mat object using the magnitude from image 1 and the phase from image 2. I created a loop that does the actual merging process for me in the code.*

4. *Display images: The program will display the hybrid image as well as the two original images.*

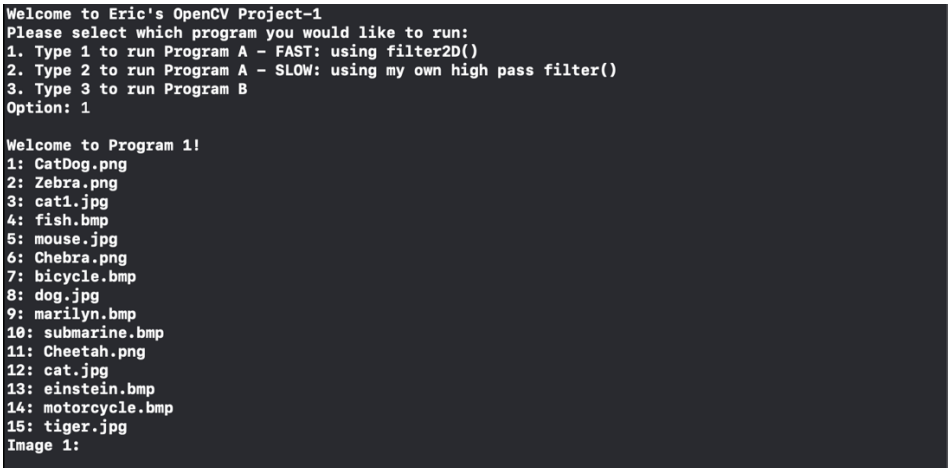
ii. Sliders control: I do not have slider control for this part of the program.

5. RESULTS

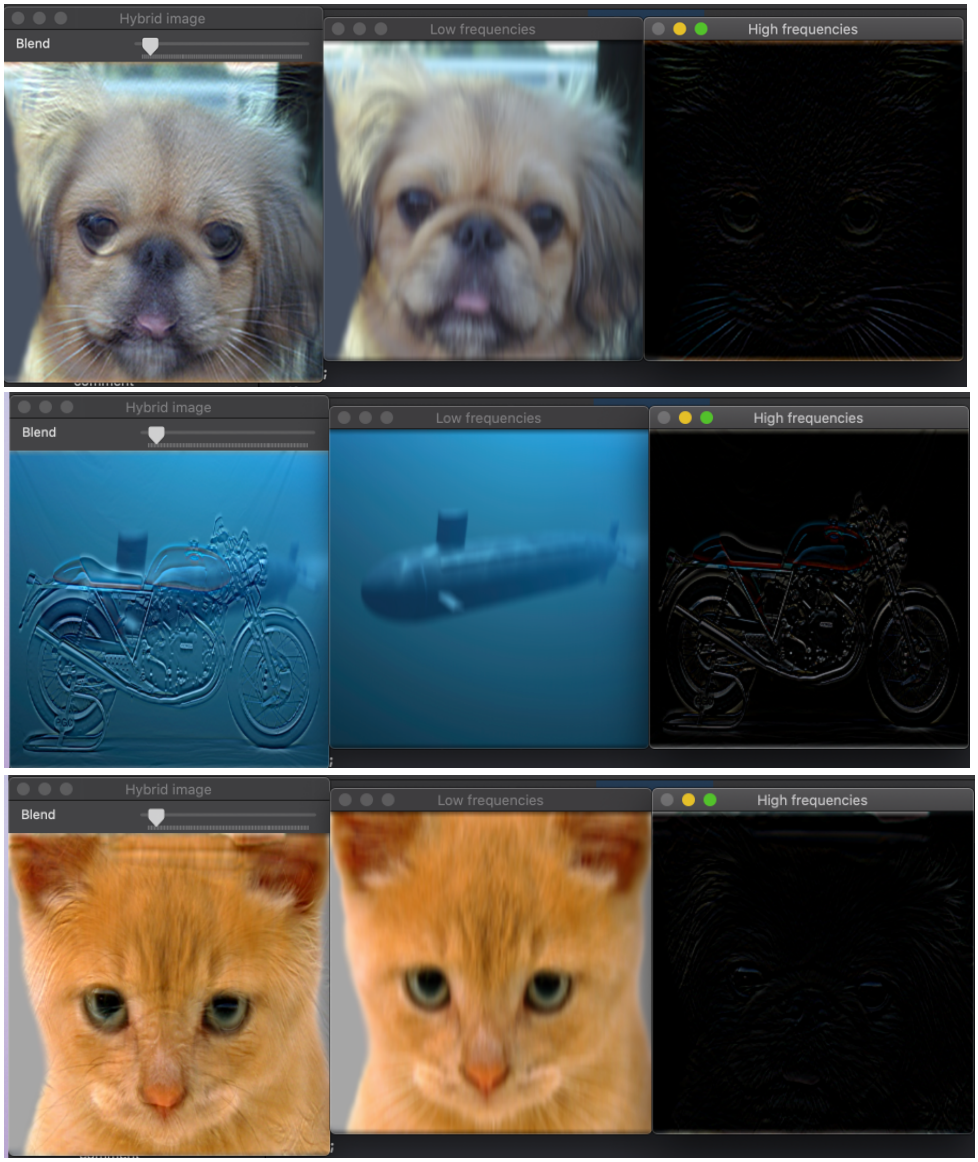
1. *User interface:*

   The following screenshot is my user interface. User is allowed to choose between 3 different modes of the program: Program A with filter2D() implementation, Program A with a high pass filter that implemented from scratch, and Program B using discrete Fourier transform (DFT). I might refer to these three modes as Mode 1, Mode 2, and Mode 3 in this summary.

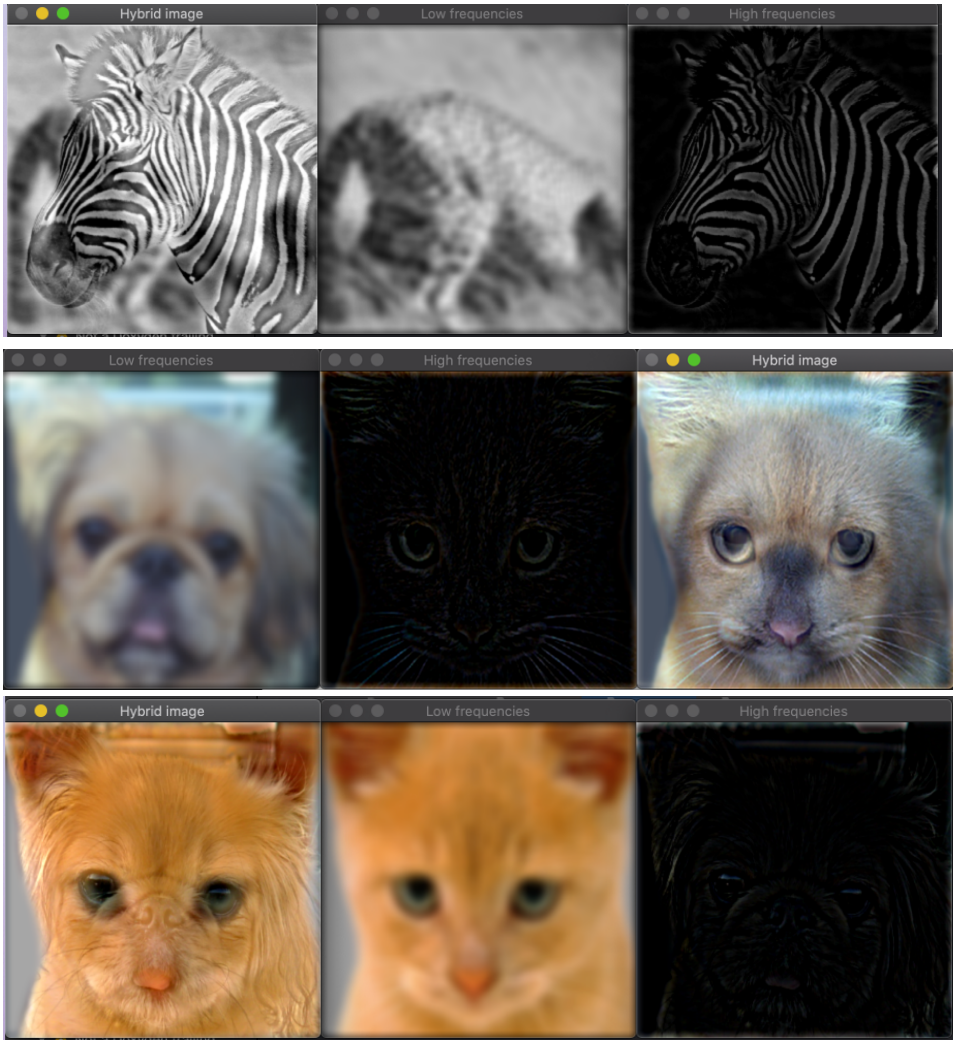
   We can notice that the Mode 1 is faster when using openCV's filter2D() function. However, when I run it with my own high pass function (Mode 2), the program is extremely slow. Thus, I decided to remove the slider control function from my Mode 2. It runs with a fixed kernel size 11 x 11.



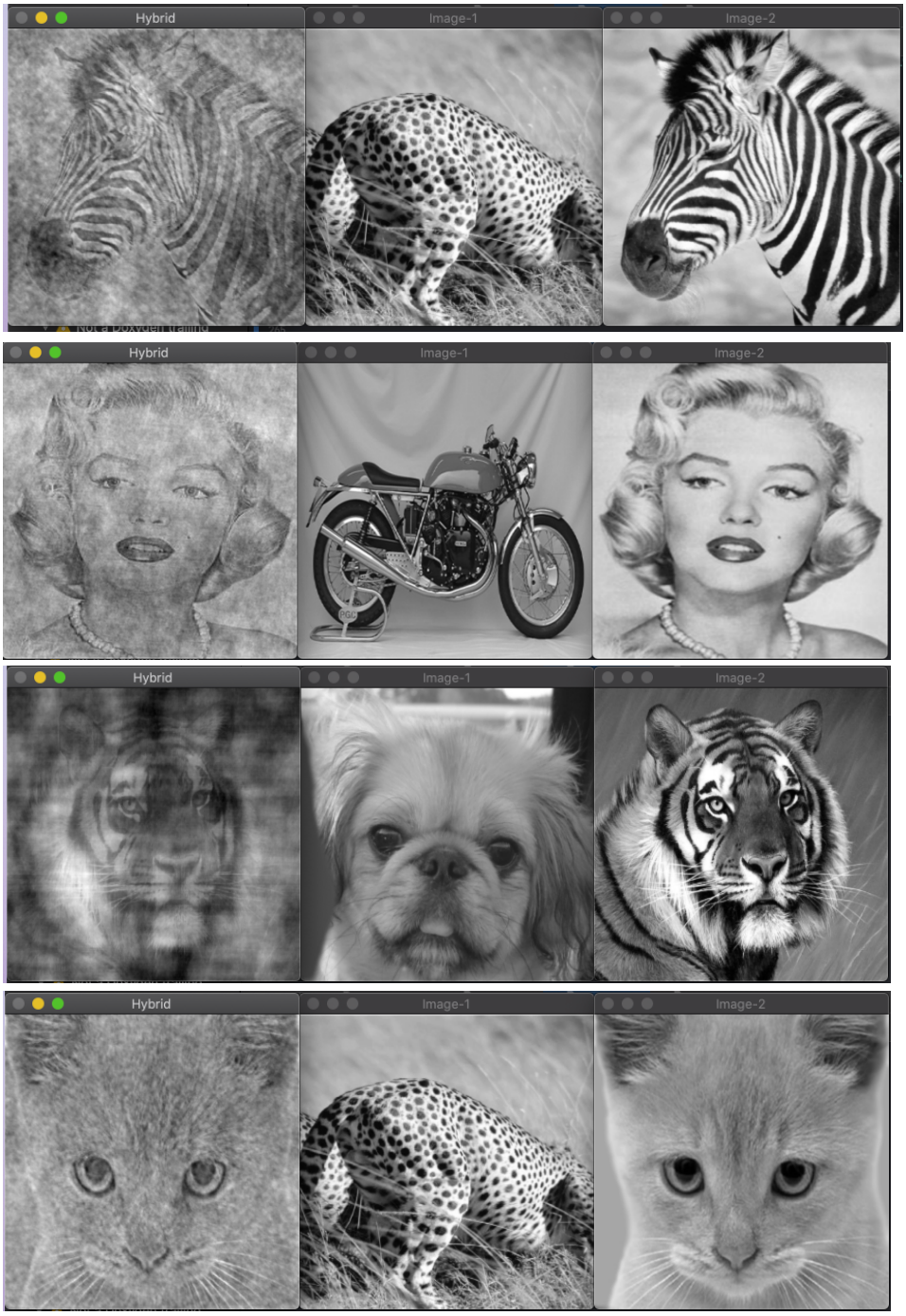
2. *Program A using filter2D()*



3. *Program A – using my own high pass filter function (slower)*



4. *Program B:*


Hybrid
Image-1
Image-2
Hybrid
Image-1
Image-2
Hybrid
Image-1
Image-2
Hybrid
Image-1
Image-2

6. DISCUSSION AND FUTURE WORK

Overall, two programs both created hybrid images that are both interesting and amusing. We can be absolutely sure that two images can be combined to create a hybrid image, with careful attention to alignment and perceptual grouping mechanisms. Indeed, when images are converted into their frequency domains, it opens a new dimension for human to explore and experiment. For me, I would love to bring more interactions into my program. For example, I want to see if there is truly a difference in looking at my hybrid image from different distances and different angles. If so, what are the relationship? I also want to do some experiments and see if I can bring my Program A together with my Program B and create some more entertaining images.

7. REFERENCE:

Oliva, Aude & Torralba, Antonio & Schyns, Philippe. (2006). Hybrid images. ACM Trans. Graph.. 25. 527-532. 10.1145/1141911.1141919.

Hutchison, D., Fusiello, A., Murino, V., & Cucchiara, R. (2012). Computer Vision - ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part III. Berlin, Heidelberg: Springer Berlin Heidelberg.

Oliva, Aude. (2013). The Art of Hybrid Images: Two for the View of One. Art & Perception. 1. 65-74. 10.1163/22134913-00002004.

Bayraktar, Ertuğrul & Yigit, Cihat & Boyraz, Pinar. (2018). A hybrid image dataset toward bridging the gap between real and simulation environments for robotics: Annotated desktop objects real and synthetic images dataset: ADORESet. Machine Vision and Applications. 10.1007/s00138-018-0966-3.