CS 342: Computer Vision

Project 1: Image Filtering and Hybrid Images

Sourav Bhattacharjee
Sawan Aich
Ramakrishna Mission Vivekananda Educational and Research Institute
Department of computer science
MSc, Big Data Analytics

Table of Contents

1. What is Image Filtering?

Image filtering (or convolution) is a fundamental image processing tool. Filters are mainly used to suppress either the high frequencies in the image, i.e. smoothing the image, or the low frequencies, i.e. enhancing or detecting edges in the image. It is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. An image can be filtered either in the frequency or in the spatial domain. In this project we will be filtering images in the spatial domain.

1.1 Examples of image filtering

Here we are using our own function to implement image filtering.

Small blur with a box filter:

Fig. 1. Original image

Fig. 2. Blurred image

Gaussian blur with a large kernel size:

Fig. 3. Original image

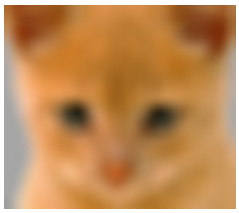

Fig. 4. Heavily blurred image

Edge detection using sobel filter:

Fig. 5. Original image

Fig. 6. Image after applying sobel filter

Implementing Laplacian filter:

Fig. 7. Original image

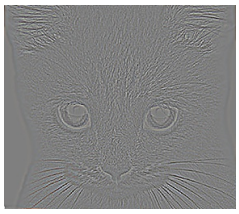

Fig. 8. High pass filter (Laplacian)

Another implementation of high pass filter:

Fig. 9. Original image

Fig. 10. High pass filter

2. What are Hybrid Images?

A hybrid image is a picture that combines the low-spatial frequencies of one picture with the high spatial frequencies of another picture producing an image with an interpretation that changes with viewing distance. Hybrid images are based on the multi scale processing of images by the human visual system and are motivated by masking studies in visual perception. These images can be used to create compelling displays in which the image appears to change as the viewing distance changes. We show that by taking into account perceptual grouping mechanisms it is possible to build compelling hybrid images with stable percepts at each distance. We show examples in which hybrid images are used to create textures that become visible only when seen up-close, to generate facial expressions whose interpretation changes with viewing distance, and to visualize changes over time within a single picture.

2.1 Examples of hybrid images

Again, our own image filtering function is used to create these hybrid images.

Fig. 11. Einstein-Marilyn hybrid image

Fig. 12. cycle-bike hybrid image

Fig. 13. bird-plane hybrid image

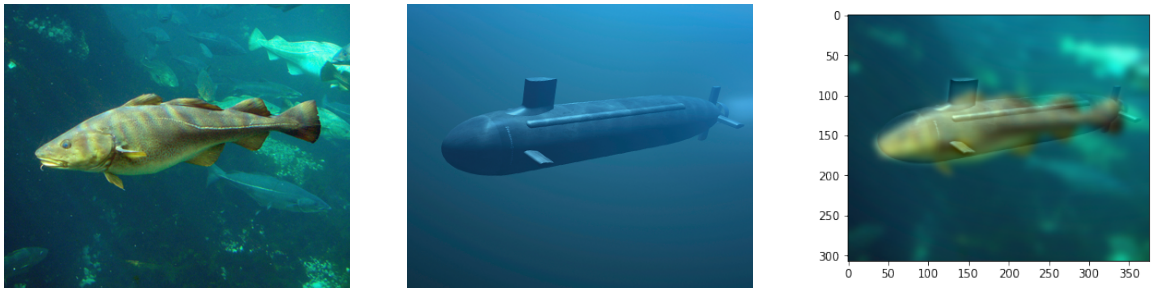

Fig. 14. fish-submarine hybrid image

3. Image filtering function algorithm

In our program, we have tried to imitate the filter2D function of the OpenCV library. The filtering is done by the "my_imfilter()" function which takes the image and the filter as the input. The image taken as the input is an RGB image, which contains three separate channels of Red, Green and Blue colours.

Now, after filtering, the original size of the image gets reduced by half the size of the filter on each side. So, we need to pad the image on all the four sides to ensure that its original size is retained. Here, we pad the three separate channels of the image with the reflected image content and stack the channels together to form the final image.

Next, we perform the correlation of the image with the filter we have taken as the input. The formula for correlation is given as follows:

$$(F * I)(x,y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} F(i,j).I(x+i,y+j)$$

This blurs the image, or in other words, removes the high frequency component of the image. The function thus returns a low-pass filtered output of the entered image.

4. Hybrid Image Formation

The hybrid image is formed by merging the low-pass filtered version of one image and high-pass filtered version of another image. The low frequency output is fetched from the "my_imfilter()" function for both the images. Then the output of the second image is subtracted from its original version to get the high frequency version of the image. Finally, the low frequency image and the high frequency are added together to produce the hybrid image.

The entire task is done by the "create_hybrid_image()" function which takes both the images and the filter as the input, and returns the low-pass first image, high-pass second image and the hybrid image.

5. Results

Here the low-pass filter is correlated with the original image to create a low frequency image of the dog:

Fig. 15. Original image

Fig. 16. Low frequency image of the dog

Then, we correlate the low-pass filter with the original image to create a low frequency image of the cat, and then subtract it from the original image to create its high frequency image.

Fig. 17. Original image

Fig. 18. High frequency image of the cat

Finally we combine the low frequency image of the dog and high frequency image of the cat to create the hybrid image.

Fig. 19. cat-dog hybrid image

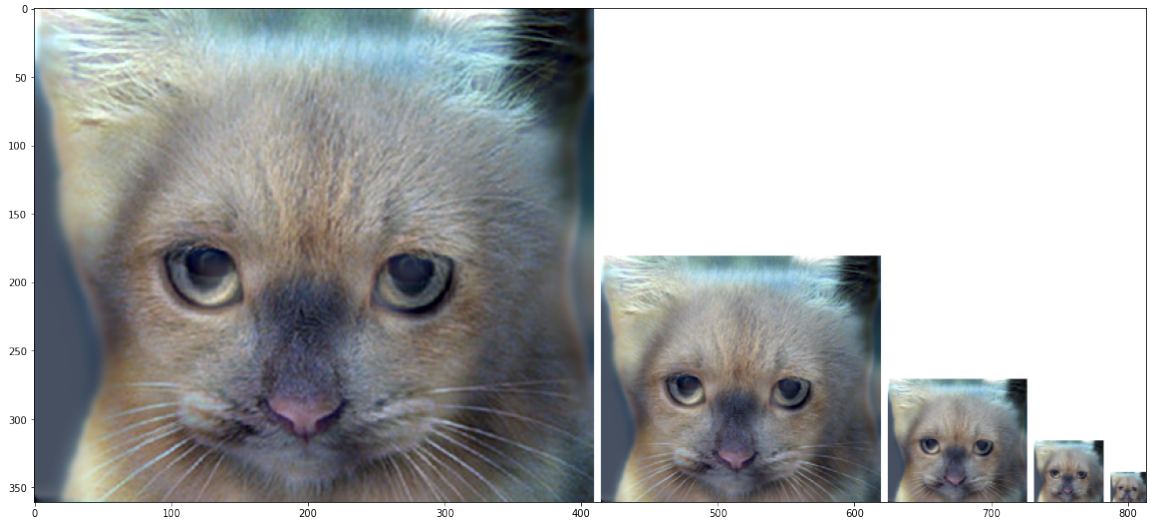

Fig. 20. As the photo gets further the cat becomes a dog!

Here we see the hybrid image in full effect!

6. Conclusion

We have discussed how filtering works on spatial domain of an image and how hybrid images are formed by combining a low frequency image with another high frequency image. Hybrid images are basically images with two interpretations that change as a function of viewing distance. From a closer distance the high frequency image component looks more prominent, and as the distance between viewer and the image increases the low frequency component becomes more prominent. This creates such interesting illusion. Despite the simplicity of the technique, the images produce very compelling surprise effects on naive observers.