Project 1 Questions

Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- Please make this document anonymous.
- On upload, **Gradescope will ask you to assign question numbers to your pages**. Making each question end with a page break after your answer is a good way to ease this process.

Questions

Q1: Explicitly describe image convolution: the input, the transformation, and the output. Why is it useful for computer vision?

A1: Image filtering is one of the widely used techniques in the computer vision as it help to extract specific information from the image help in many application. Convolution , beside correlation, is the mathematical representation of filtering as it is the process used to filter some image with the kernel. The input of the convolution , in the scope of computer vision, is the image and the kernel. The transformation and the output are held as the upcoming equations.

$$x(t) * h(t) = y(t)$$

$$X(f)H(f) = Y(f)$$

Q2: What is the difference between convolution and correlation? Construct a scenario which produces a different output between both operations.

Please use scipy.ndimage.convolve and scipy.ndimage.correlate to experiment!

A2: The mathematical difference appear in these two equations :

$$f * g(x) = \int f(y)g(x-y)dy$$

$$f \circ g(x) = \int f^*(y)g(x+y)dy = f^*(y) * g(-y)$$

On the other hand , indeed they do the same process but with reversed kernel , as if the kernel is symmetric they give the same results

Q3: What is the difference between a high pass filter and a low pass filter in how they are constructed, and what they do to the image? Please provide example kernels and output images.

A3: The digital images , as we treat them that way in the computer, contains values represent the intensity of the image in this specific pixel. So, the difference between the value and its neighbors determine the frequency as if the image is homogeneous the colour variations are rare so a lot of low frequencies in the images. On the other hand, if a lot of variations in the values of pixels , there are a lot of high frequencies in this image. So the low pass filter is constructed to capture the low frequencies so it blurs the images :

1/9 1/9 1/9
1/9 1/9 1/9
1/9 1/9 1/9 and the high pass filter is constructed to capture the high frequencies so it sharpen the images :

-1/9 -1/9 -1/9
-1/9 -8/9 -1/9
-1/9 -1/9 -1/9

Q4: How does computation time vary with filter sizes from 3×3 to 15×15 (for all odd and square sizes), and with image sizes from 0.25 MPix to 8 MPix (choose your own intervals)? Measure both using $scipy.ndimage.convolve$ or $scipy.ndimage.correlate$ to produce a matrix of values. Use the $skimage.transform$ module to vary the size of an image. Use an appropriate charting function to plot your matrix of results, such as $Axes3D.scatter$ or $Axes3D.plot\_surface$.

Do the results match your expectation given the number of multiply and add operations in convolution?

Image: RISDance.jpg (in the .tex directory).

A4: Your answer here.