Project 1 Questions

Questions

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

A1: Convolution is the core operation in image and signal processing.It is mainly used to apply various filter types over images.Its importance comes from the fact that it has direct connection with the frequency domain of image ,as the convolution operation in the spacial domain can be converted into multiplication in the frequency domain and vice versa.The input of the convolution is two images with any size. the transformation itself occurs by the following equation.

$$(f * g)[n] = \sum_{m \to \infty}^{\infty} f[m]g[n-m] \tag{1}$$

The output of convolution is a single value for each iteration in the summation , so for an image with convolutional filter , each new pixel in the filtered image will be affected by the pixels next to it.It is also worth mentioning that the size of the filtered image is slightly less than the original one due to the pixels in the borders,hence we need padding to make the output filter with the same dimension as the original one.

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: They are basically the same operation ,but the convolution flips the filter firs before performing the multiplication and summation.in other words,correlation and convolution can be used interchangeably with taking into consideration the filter used.

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 low pass filter mainly focuses on leaving non-abrupt changes in the image and the noise can be removed using this. The main methodology of low pass filtering is to mean each pixel with its neighboring pixels this vanishes the effect of non-smooth changes.There are many kinds of low pass filters. There are box filters, Gaussian filters, and more.
An example of a low-pass filter kernel is this:

$$k = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

High-filters on the other hands does the exact opposite. It keeps only the abrupt changes in the images and discard any smooth changes. So it keeps any noise, for example. Edge detection is considered high pass filtering as it keeps the sudden changes in the pixel values at the edges.
An example of a high-pass filter that detects vertical edges:

$$k = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

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.