COMP9517

Lab 2, S1 2018

This lab presents a revision of important concepts from week 2 and 3 lectures. Most questions require you to use OpenCV, an open source software package that is widely used in the field. OpenCV v2.4.9.1 is installed on the lab machines and accessible through vlab environment. You are expected to use vlab during the lab sessions.

For information, the latest version of OpenCV is downloadable from https://opencv.org/

All questions should be attempted during the lab hour.

The last question (Question 4) is assessable IN THE LAB. Make sure to show your answer to your tutor before leaving the lab. It will not be assessed later on outside the lab in person or by email.

DATA SAMPLES can be downloaded from the class website.

IMAGE FILTERING

Image filtering is a neighbourhood operation for modifying or enhancing an image, that can emphasise certain features or remove certain unwanted features. In computer vision, filtering is mainly used to eliminate undesirable characteristics that may be formed by random variations in the intensity or illumination. Image filtering is usually implemented as a convolution.

Convolution is a mathematical operation which takes as input two functions (say f and g) and produces an output function (say h) which typically is a modified version of one of the original functions. In our case, f is the input image, g is the convolution mask/kernel, h is the filtered version of f and convolution creates weighted sums of the image pixels. Convolution in the spatial domain corresponds to multiplication in the frequency domain. Therefore, convolution using large filters (expensive in spatial domain) can be more efficiently implemented using fast Fourier transform (FFT). In computer vision most of the filter sizes are small and filtering is implemented as convolution in the spatial domain (as FFT provides little or no benefit for small filter sizes).

Smoothing Filters (Low-pass Filters)

Noise (random variations in intensity) removal from images is implemented using Image filtering. Salt and pepper noise, impulse noise and gaussian noise are some of the commonly observed types of noise. Different types of filters are suited for different types of noise. A smoothing filter is often called a low-pass filter as it allows low frequency components of the image to pass though (regions with similar intensity values) but stops high frequency components from passing through (edges or noise). Shown below are the results of applying different filters on an image corrupted by salt and pepper noise. It can be observed that the median filter produces the best result in this case.

Original Image

Mean filter (size:3*3)

Mean filter (size:5*5)

Noisy Image

Median filter (size: 3*3)

Gaussian filter (size:3*3, σ =1.5

QUESTION 1: Implement a function which can perform convolution operation given an image and a convolution mask/kernel.

Note:

1. Standard functions are available, but it is advisable to implement you own function to get a better understanding.
2. While convolving at the boundary of an image, a part of the kernel is off the image boundary, make sure to use appropriate ***padding*** methods to avoid errors.

QUESTION 2: Implement a mean filter, median filter and a gaussian filter. Perform noise removal on the images provided in the course website. Try with different filters and kernel sizes, observe the variations and find the best filter and kernel size for the noisy images.

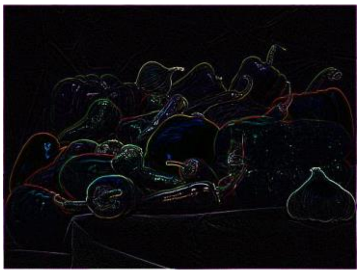
Image Sharpening using High-pass filters

Image sharpening is used to highlight fine details in an image or to enhance features that are blurred. High pass filters allow high frequency components such as edges and noise to pass through, while blocking low frequency components. Given below are examples of high-pass filter kernels:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

To sharpen an image, add the original image and the high-pass filtered image and the result will be a new image with enhanced edges (the same result can be produced by incrementing the centre value of the kernel by 1). Shown below are the results of applying a high-pass filter and the resulting sharpened image.


Original Image


High-pass filtered Image


Sharpened Image

QUESTION 3: Implement a high-pass filter and perform Image sharpening on the images provided in the course website. (A high pass filtered image can also be obtained by subtracting the low pass filtered image from the original image).

Hybrid Images

High pass filters and low pass filters are the two main types of image filters. Images filtered with a low pass filter are smoothened, while those filtered with high pass filters are sharpened. If a high-pass filtered image is combined with a low pass filtered image, a hybrid image [3] is formed which changes its content as the distance between the image and the viewer changes. Shown below are two sample images, their high-pass filtered, and low-pass filtered versions and the resulting hybrid image (Pictures from [4] and [5]).


Original Image 1


Original Image 2

High-pass filtered Image

Low-pass filtered Image

Hybrid Image

QUESTION 4: Read two images, perform high pass filtering on one and low pass filtering on the other. Create a hybrid image by adding the two results.
Hint: You may re-use appropriate filters already implemented for Questions 2 and 3.

Note:

1. It is sometimes difficult to visually see the high pass filtered images as they have zero mean and negative values, this can be overcome by adding a small value to the image pixels (say 0.5).
2. To get a better view of the two visualizations without having to walk away from the screen, it will suffice to down sample the image.

REFERENCES

[1]. http://radio.feld.cvut.cz/matlab/toolbox/images/linfilt2.html

[2]. http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter4.pdf

[3]. Aude Oliva, Antonio Torralba and Philippe G. Schyns (2006). "Hybrid images" (PDF). ACM Transactions on Graphics (SIGGRAPH 2006 issue). 25 (3): 527–532.

[4]. https://www.pexels.com/search/dog/

[5]. https://www.pexels.com/search/cat/