ECE 471 - Computer Vision
Assignment 1

Image Processing

Arshi Annafi
V00[REDACTED]

University of Victoria. Fall 2019.

Contents

Background

Image processing is a method to perform some operations [i.e. apply algorithms] on an image in order to get an enhanced image or to extract some useful information from [the image] [1].

In this assignment, image of a pear will be enhanced by sharpening the image. This will be accomplished using an approximation of Laplacian of Gaussian filter i.e. sharpening filter. Laplacian of Gaussian is approximated using an impulse and a Gaussian kernel; discussed in details in the report.

Note that the scope of this assignment is limited to greyscale images only. I.e. colored images are converted to greyscale image before sharpening.

Gaussian Kernel

The Gaussian kernel was generated using the following formula [2]:

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

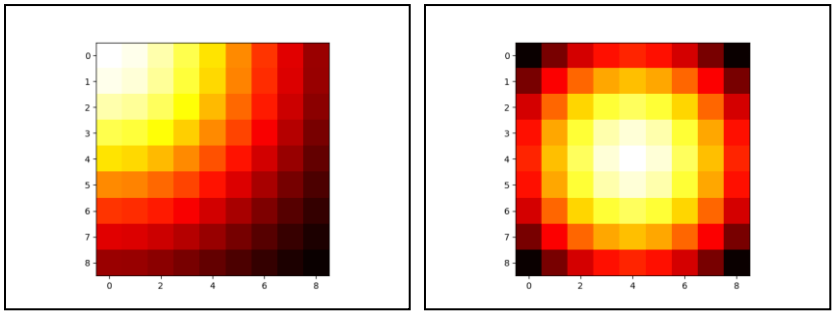
Initially, the $x^2 + y^2$ term was implemented as following:

```
For i in range(kernel_size):
      For j in range(kernel_size):
            x2_plus_y2 = i**2 + j**2
            ...
```

This generated a Gaussian curve with its peak at coordinate (0,0) as seen in figure 1 a. Then the above code was updated to include an offset:

```
...
offset = np.floor(kernel_size /2)
x2_plus_y2 = (i-offset)**2 + (j-offset)**2
...
```

to bring the peak of the Gaussian curve to the center of the kernel as seen in figure 1 b. This method of centering the peak of the Gaussian curve was inspired by [3].



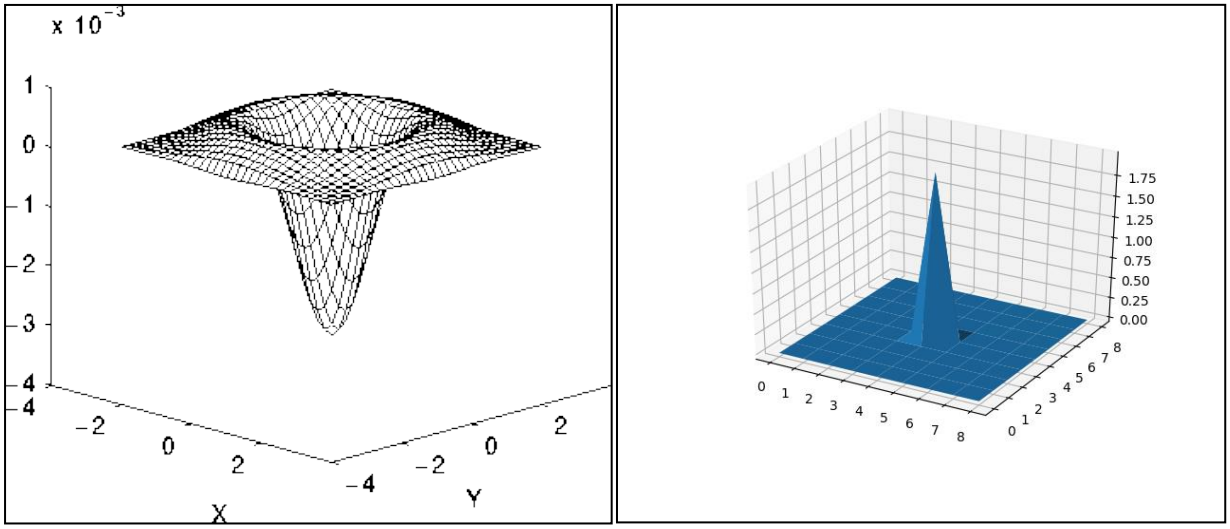
(a) (b)

Figure 1 - Heat map of the 9x9 Gaussian kernel

Finally, the kernel was normalised since the sum of the kernel should equal 1.

Laplacian of Gaussian (Sharpening Filter)

Image is sharpened using Laplacian of Gaussian (LoG) filter. Theoretically, the LoG looks like figure 2 a. And figure 2 b. is what the LoG (inverted) looks like in a discreet domain; kernel size 9x9.



(a) Theoretical LoG [4] (b) LoG generated by the code

Figure 2 - Laplacian of Gaussian (LoG)

Sharpening filter was achieved i.e. LoG was approximated by subtracting the Gaussian kernel (figure 1 b) from the unit impulse kernel (figure 3).

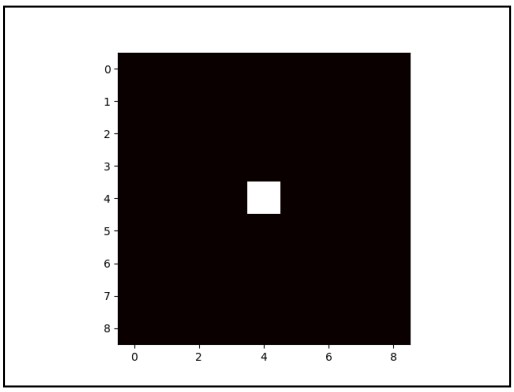


Figure 3 - A 9x9 unit impulse kernel.

Padding

Before applying the attained sharpening kernel, the input image was padded. Padding is necessary because convoluting the kernel with the image will reduce the size of the image. The padding scheme that was used is "reflect" as recommended. This padding scheme simply reflects the boundary pixels along the axis and onto the newly added pixels around the image.

Convolution

Formally, convolution [2] is:

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

Procedurally, as see in figure 4, the kernel is flipped horizontally and vertically. Then for each local neighborhood of the image (sliding window), pixel values from the image are multiplied with the respective values of the kernel. Then all values are added to produce the resulting pixel value of the output image.

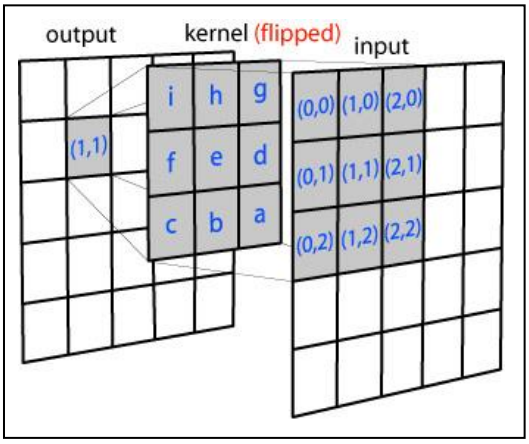


Figure 4 - Convolution of the kernel with the image [5]

Clipping

After performing the convolution, the image is essentially sharpened. Before finalising and saving the image, it is necessary to clip all values that are not in the desired range of values. I.e. change all pixel values that are smaller than zero and greater than 255 to zero and 255 respectively.

Result

This section shows and discusses the outcome of the assignment. Figure 5 a. is the input image and figure b. is the resulting sharpened image. The kernel was a 9x9 kernel and σ of the Gaussian kernel was 5.

(a) Unsharpened (b) Sharpened

Figure 5 - Image of a pear before and after sharpening.

Works Cited

[1] University of Tartu, “Digital Image Processing,” [Online]. Available: https://sisu.ut.ee/imageprocessing/book/1. [Accessed 23 January 2019].

[2] A. Dash, “Lecture 2: Image Processing,” Victoria, 2019.

[3] Geek Bit of Everything, “Gaussian filter implementation in Matlab for smoothing images (Image Processing Tutorials),” YouTube, 4 December 2017. [Online]. Available: https://www.youtube.com/watch?v=LZRiMS0hcX4. [Accessed January 2019].

[4] R. Fisher, S. Perkins, A. Walker and E. Wolfart, “Laplacian/Laplacian of Gaussian,” 2003. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm. [Accessed 23 January 2019].

[5] Bert, “How is Gaussian Blur Implemented?,” Stack Exchange, 5 August 2015. [Online]. Available: https://computergraphics.stackexchange.com/questions/39/how-is-gaussian-blur-implemented. [Accessed 22 January 2019].