INDIAN INSTITUTE OF TECHNOLOGY

KHARAGPUR

DIGITAL IMAGE PROCESSING LABORATORY

A REPORT ON

EXPERIMENT 03

Spatial Filtering

15.02.2021

Group No. 16

Name:	Gambhira Sirish	Shubham Sahoo
Roll No:	**17EC35009**	**17EC35023**

DEPT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

VISUAL INFORMATION AND EMBEDDED SYSTEMS

Table of Contents

Introduction

Spatial Filtering is a technique used directly on pixels of image. Mask is usually considered to be added in size so that it has a specific center pixel. This mask is moved on the image such that the center of the mask traverses all image pixels. Image filtering can be used to:

1. Remove noise
2. Sharpen contrast
3. Highlight contours
4. Detect edges

They are of two types:

1. Linear filters
2. Non-linear filters

Linear filters: The pixels in the output image are obtained as a sum of product of filter (mask) coefficients with the pixel intensities of input image. Eg: Mean filter, Gaussian filter, etc..

Non-linear filters: The pixels in the output image are obtained by some conditioning on input pixel neighborhood. They do not explicitly use the sum of products' manners as in case of linear filters.. Eg.: median filter.

Based on their effect, spatial filters are classified into:

1. Smoothing spatial filter/ Lowpass filter
2. Sharpening spatial filter/ High Pass filter

Smoothing filters are used to remove noise from the image, whereas sharpening filters are used to extract edges from the image.

Algorithms

1. Mean filter

1. Kernel of size d X d is created where each coefficient is 1/ (d * d).
2. This kernel is convoluted over the input image and the resultant output image is mean-filtered.

2. Median filter

1. For each pixel in the output image, a corresponding neighborhood of the input image pixel is selected.
2. The values in this neighborhood are sorted and the median value is returned.
3. This is applied for all pixels of the output image, thus median-filtering is achieved.

3. Gaussian filter

1. Kernel of size d X d is created, where kernel[i,j] is given by [3]:

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

2. The kernel coefficients are normalised to 1, since it is a low-pass filter
3. This kernel is convoluted over the input image to produce the output image.

4. Laplacian

1. Kernel of size d X d is create where all the pixels (except the centre) are 1 and the centre pixel is - (d * d - 1)[4].

5. Laplacian of Gaussian

1. Laplacian of gaussian is obtained by first applying a gaussian smoothing followed by a laplacian filter on top of it.
2. We also experimented with LoG filters by using predefined kernels and using formula based kernels.

6. Sobel (Horizontal, Vertical, Diagonal)

1. Sobel filters are defined for kernels of size 3. Hence any kernel size other than 3 is rejected for this filter.
2. For the horizontal and vertical case, we obtained the filtered image by convolving with the pre-defined sobel kernels.
3. For diagonal case, we obtain the filtered image by first convolving the image with diagonal kernels and then take the resultant of both images.

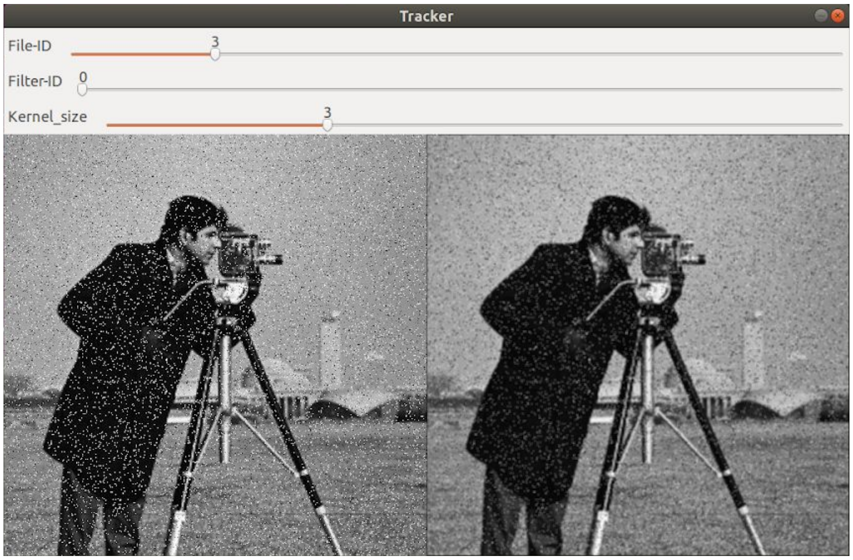
7. Prewitt

1. Prewitt filters use the same equations as the sobel filters, except that they don't place any emphasis on pixels that are closer to the center.
2. We obtain the horizontal and vertical filtered images from the Prewitt kernels and take their resultant as the filtered image.

Results

Mean filtering

Salt and pepper noise

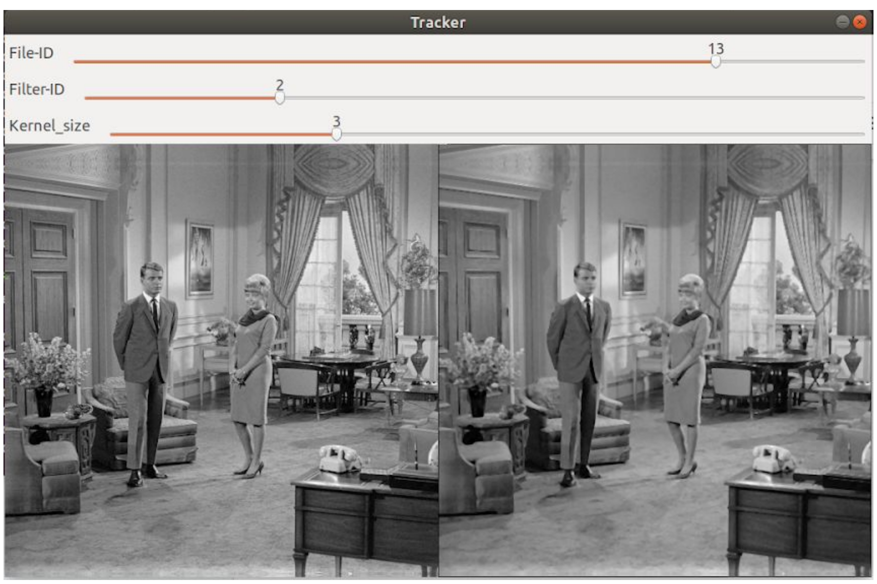


Gaussian Noise

Median Filtering

Salt and pepper noise

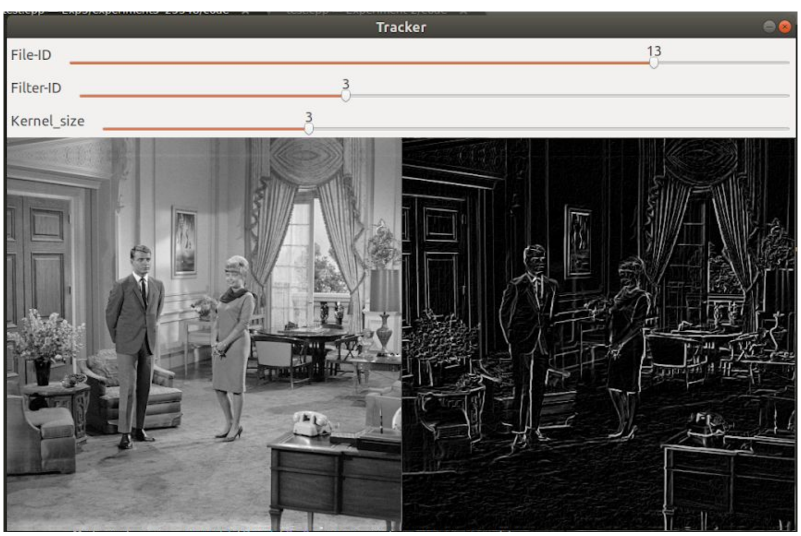
Noiseless



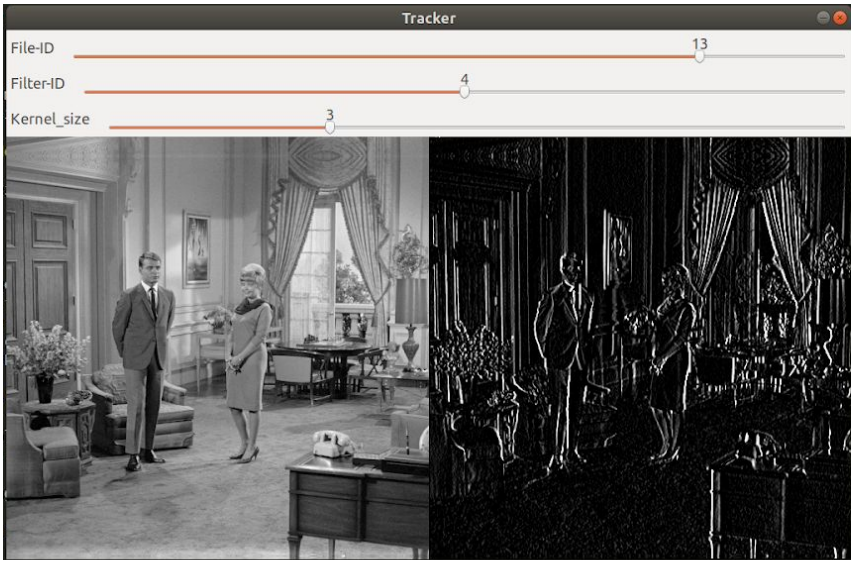
Gaussian filtering

Gaussian noise

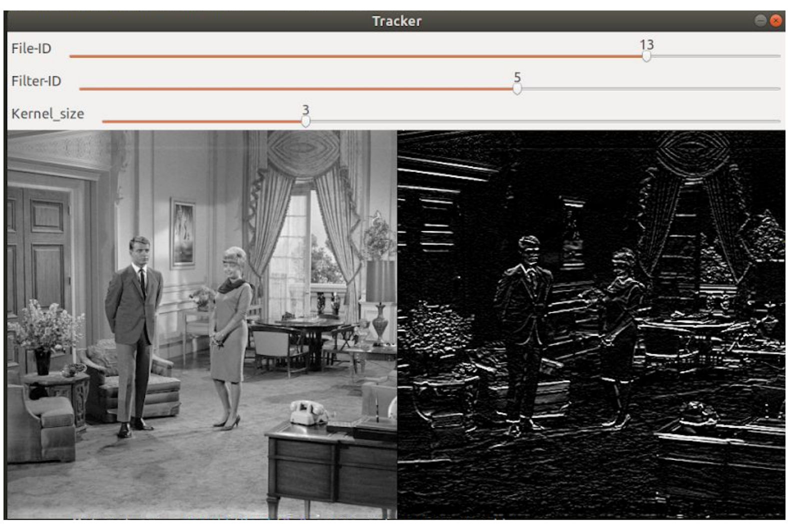
Prewitt filter



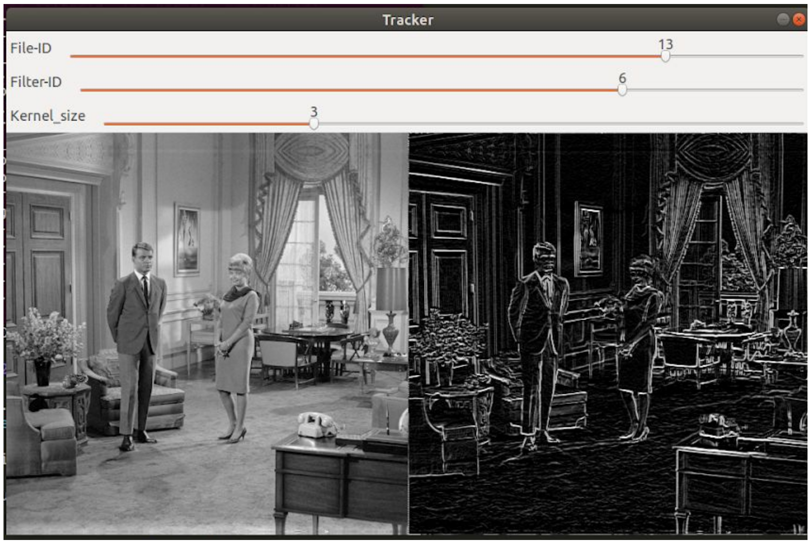
Sobel - Horizontal filter



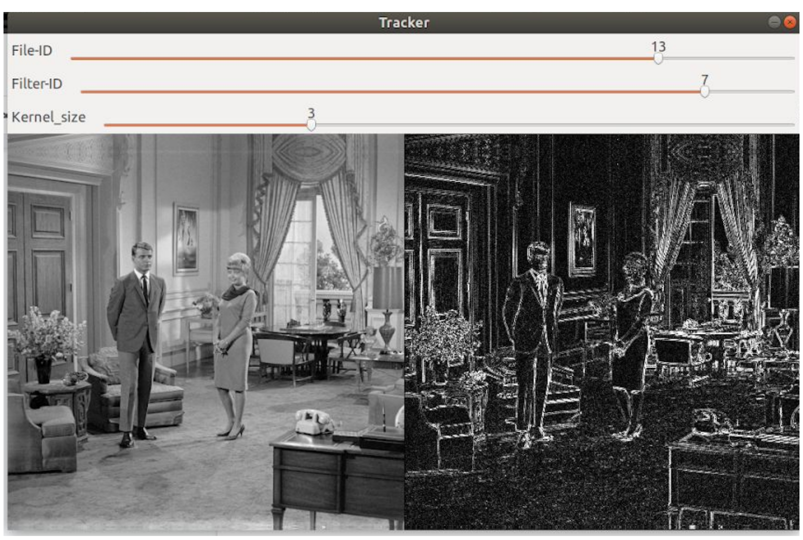
Sobel - vertical filter



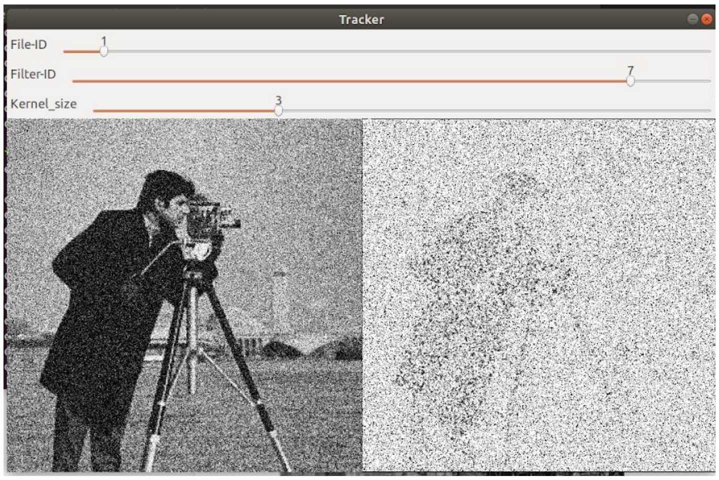
Sobel - diagonal filter



Laplacian

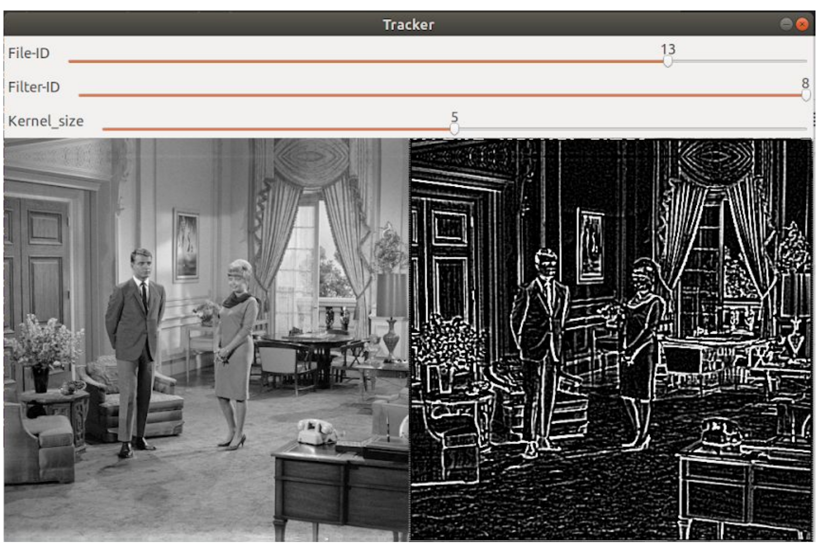


(Laplacian filter on gaussian noisy image)

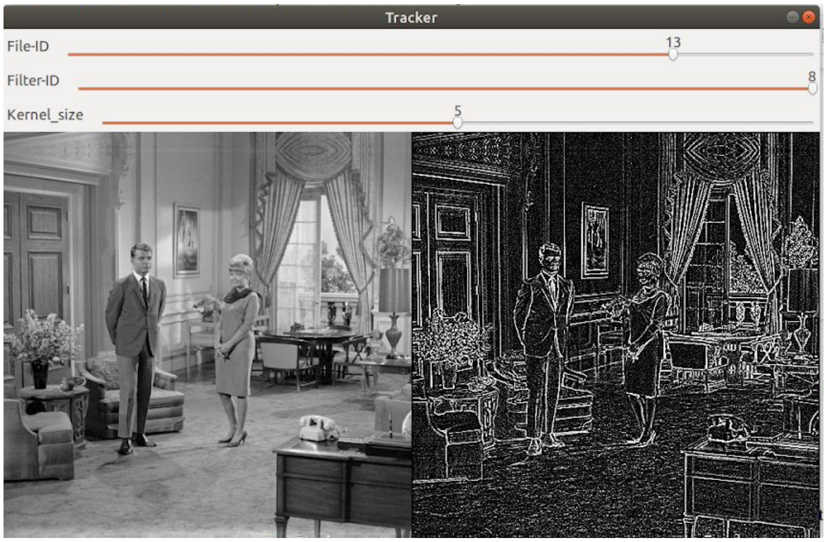


Laplacian of Gaussian

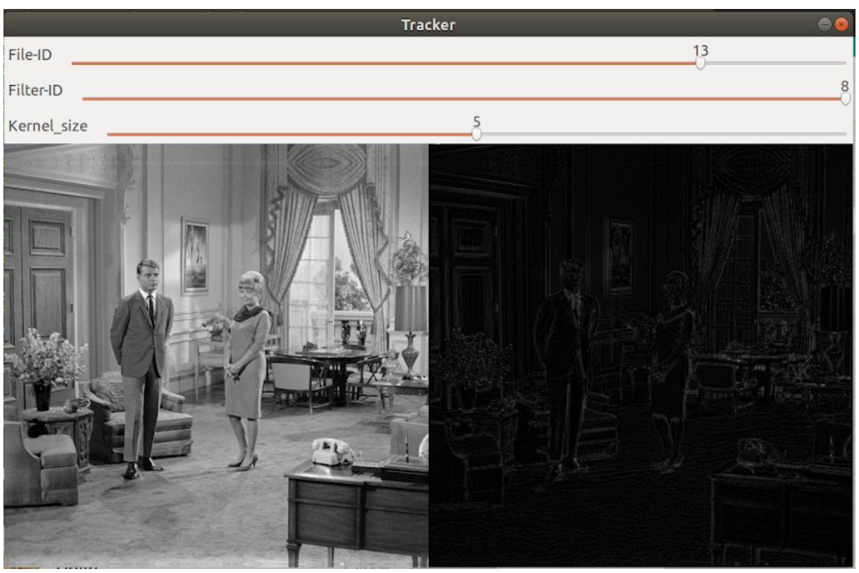
(Gaussian smoothing followed by laplacian - cascaded)



(Hardcoded Kernels)



(Formula based)



Analysis

- The mean-filtering is not able to remove either salt and pepper noise/ gaussian noise completely. It only reduces the noise intensities, however they are still visible.
- The median-filtering removes salt-pepper noise clearly as expected. However, when applied to real-images, we can lose important information in images such as earrings (which are identified as noise).
- We used a gaussian formula [3] to generate the kernels. We got erroneous results when using the kernel directly, but later realised that the sum of coefficients should be 1 for LPF.
- The standard deviation in the case of Gaussian kernel is determined by the formula σ = kernel_size/6 [1].
- Prewitt filter is a simple high-pass filter whose coefficients are derived from the standard derivative of a function.
- Sobel filter is similar to Prewitt, but in this we give more weights to the pixels near the center.

- For laplacian, we used predefined kernels given by:

−1	−1	−1
−1	8	−1
−1	−1	−1

and extended this to kernel sizes 5, 7, 9 [4].

- We understand this discrete approximation is valid only for lower size kernels, however, we found that laplacian is usually replaced by laplacian of gaussian for higher kernel sizes.
- The laplacian filter is sensitive to noise and as expected with increase in kernel size, more noise signal is picked up, which can be observed in the results.
- For LoG, we had three options: either following gaussian smoothing followed by laplacian filter, using pre-defined kernels[5], or discrete approximation directly from the formula.

- We observed that gaussian smoothing followed by laplacian filter is identifying similar edges to sobel and prewitt filters. The predefined kernels are identifying even subtle edges also, which may not be intended
- The formula based LoG filter is a good approximation only for larger kernel sizes.
- For sobel and prewitt, we only implemented kernels of size 3 as we couldn't gather resources with explanations for higher order. It is written to solve a differential equation and then find the values.

Contributions:

Sirish: Mean, Median, Gaussian, Laplacian, LoG, Trackbar

Shubham: Sobel (H,V,D), Prewitt, Part of Laplacian, Add-on to Gaussian

References

1)http://sdeuoc.ac.in/sites/default/files/sde_videos/Digital%20Image%20Processing%203rd%20ed.%20-%20R.%20Gonzalez%2C%20R.%20Woods-ilovepdf-compressed.pdf

2)https://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter5.pdf

3)https://en.wikipedia.org/wiki/Gaussian_filter#:~:text=In%20electronics%20and%20signal%20processing,as%20it%20has%20infinite%20support).

4)https://stackoverflow.com/questions/19422029/how-to-calculate-a-laplacian-mask-or-any-size

5)https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm