CS250B Lab2 Report

Fu-Cheng Hsieh

1 Horizontal Filter

Figure Fig 1 is a rough sketch of my implementation for the horizontal filter. First, I duplicate the input stream from the row buffer in the middle and send the value to a streaming buffer, then duplicate the value and send it to three buffers. By sending the values to three buffers, the first, second, and the third, with controlling how the convolution module consumes data, we can simulate the iteration of convolving filter to horizontal pixels.

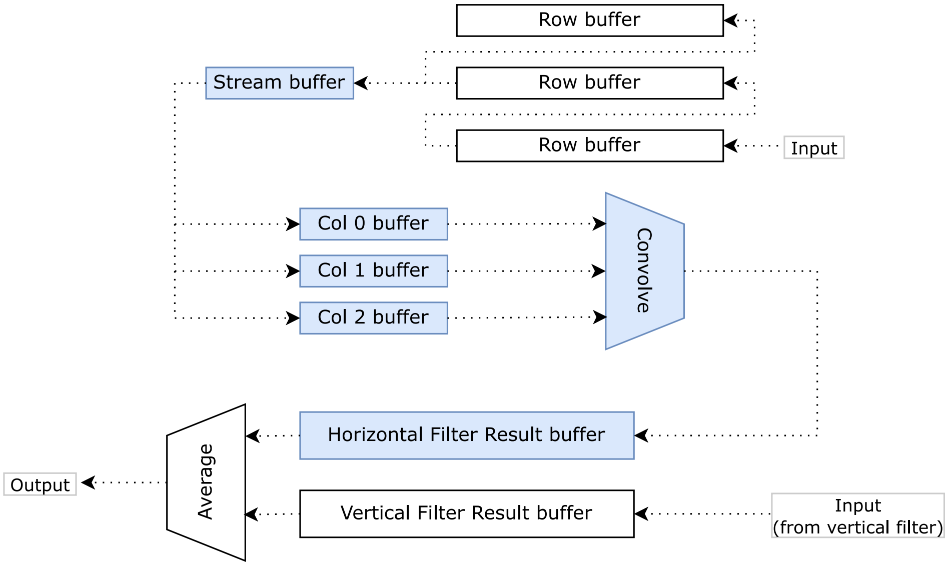
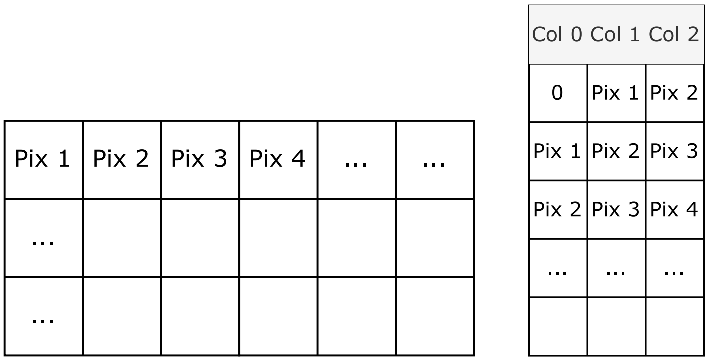


Figure 1: Schematic of the horizontal filter



Pix 1	Pix 2	Pix 3	Pix 4	...	...
...					
...					

(a) An Example image

Col 0	Col 1	Col 2
0	Pix 1	Pix 2
Pix 1	Pix 2	Pix 3
Pix 2	Pix 3	Pix 4
...	...	...

(b) Sequence of pixels

Figure 2: An example image and the input sequence of column buffers

For example, if the image's layout looks like the one in Fig 2a, then the column buffers will have an input sequence that looks like in Fig 2b.

I also add an indicator of how many pixels we have output from the convolution module so that when we reach the end of a row $(ind \mod 512 == 511)$ or is a start of a row $(ind \mod 512 == 0)$, we can change the third or the first col to 0 so that the properties of horizontal filter convolution remains.

2 Vertical Filter

The schematic of the middle case is as shown in Fig 3, and the process illustration is as shown in Fig 4.

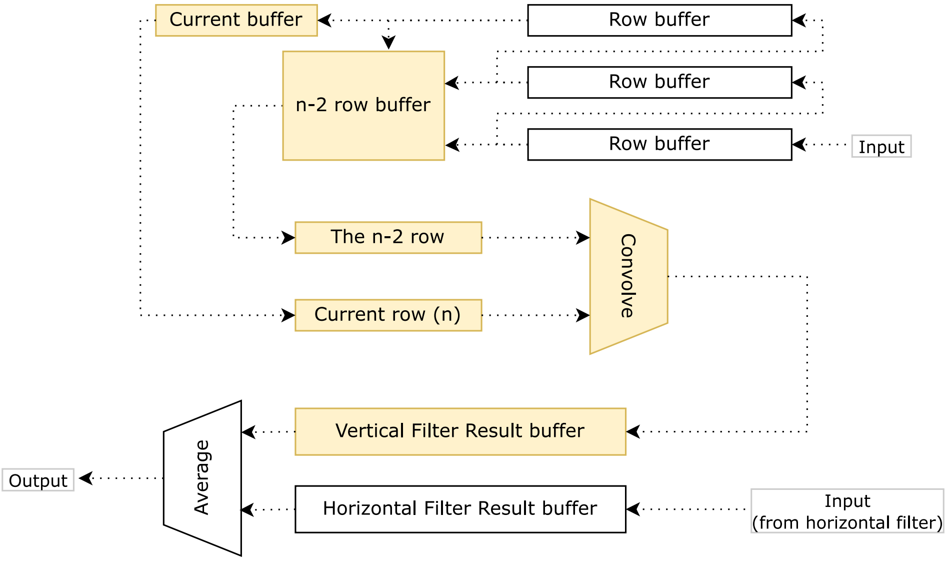


Figure 3: Schematic of the vertical filter (the middle case)

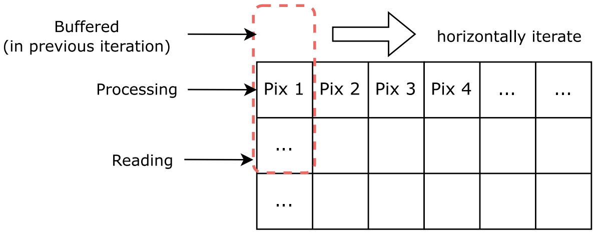


Figure 4: Illustration of vertical filter processing

I separate the question into three cases for the vertical filter: first row, last row, and rows in the middle.

2.1 The first row

Starting with pixel id 0 to 511, we need to wait until we are reading the second row (512 to 1023) to process the vertical filter convolution. Here I use a counter so that we wait until we are starting to read pixel #512, then start the vertical filter processing.

2.2 The middle case

Starting from pixel 1024 to $512 * 255 - 1$. Since we have cached the row before last, when we are reading a row, we can start outputting the convolution result which should be placed in the previous row.

2.3 The last row

Starting with pixel id $512 * 255$ to $512 * 256 - 1$. Since we have saved the values of the row before last to a buffer, we can now fetch the data from this buffer and convolve it, which is equal to outputting this value with a negative sign since vertical filter convolution is third row − first row, and the third row is 0.

3 Result

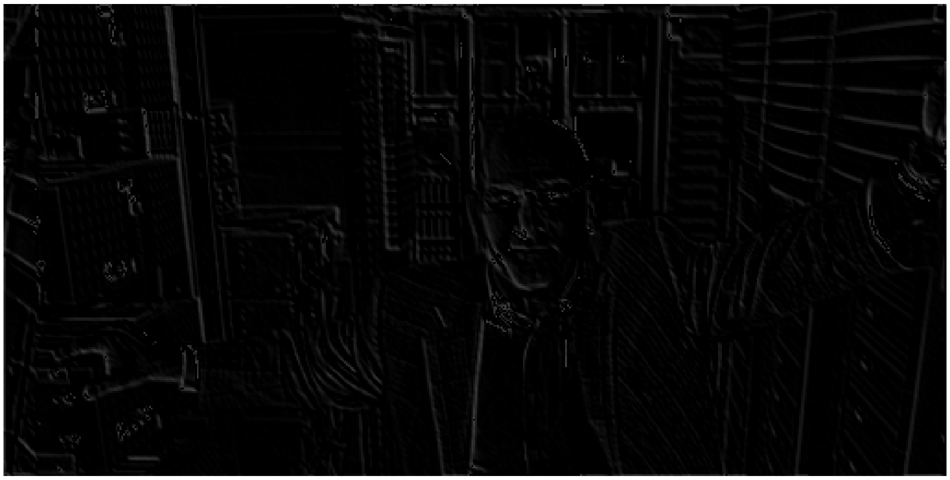
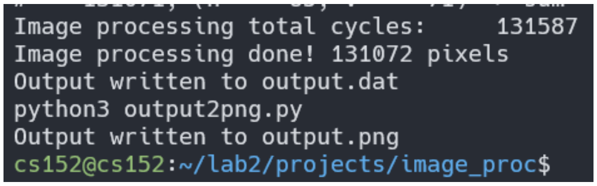


Figure 5: The result image



```
Image processing total cycles:     131587
Image processing done! 131072 pixels
Output written to output.dat
python3 output2png.py
Output written to output.png
cs152@cs152:~/lab2/projects/image_proc$
```

Figure 6: Image processing cycles

The output image is as shown in Fig 5. I set the negative values to 0 since I have no idea how to handle this case, so it seems to be a little bit dim. The total number of cycles is 131587, as shown in Fig 6.