DATA.ML.300 Computer Vision Exercise 1

Toivo Wuoti

January 2022

1

1.1 a)

Cartesian coordinates

$$
\begin{aligned}
x_1 &= (2, -1) \\
x_2 &= (1, -2) \\
x_3 &= (1, 1) \\
x_4 &= (-1, 0)
\end{aligned}
$$

And their corresponding homogenous can be made by adding 1 as the last element, thus them being

$$
\begin{aligned}
x_1 &= \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \\
x_2 &= \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \\
x_3 &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\
x_4 &= \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

1.2 b)

The line l_1 can be calculated such that

$$l_1 = x_1 \times x_2 = \begin{bmatrix} 1 \\ -1 \\ -3 \end{bmatrix}$$

And the line for l_2 can be calculated similarily

$$l_2 = x_3 \times x_4 = \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix}$$

1.3 c)

Finally we can get the intersection point x by calculating the cross product of l_1 and l_2

$$x = l_1 \times l_2 = \begin{bmatrix} -7 \\ -4 \\ -1 \end{bmatrix}$$

And from these homgenous coordinates we can get the cartesian coordinates by dividing the vector with the last value, so that

$$x = (7, 4)$$

Computer Vision Exercise 1

1

a)

```
x1 = [2 -1 1].'
```

```
x1 = 3×1
     2
    -1
     1
```

```
x2 = [1 -2 1].'
```

```
x2 = 3×1
     1
    -2
     1
```

```
x3 = [1 1 1].'
```

```
x3 = 3×1
     1
     1
     1
```

```
x4 = [-1 0 1].'
```

```
x4 = 3×1
    -1
     0
     1
```

b)

```
l1 = cross(x1, x2)
```

```
l1 = 3×1
     1
    -1
    -3
```

```
l2 = cross(x3, x4)
```

```
l2 = 3×1
     1
    -2
     1
```

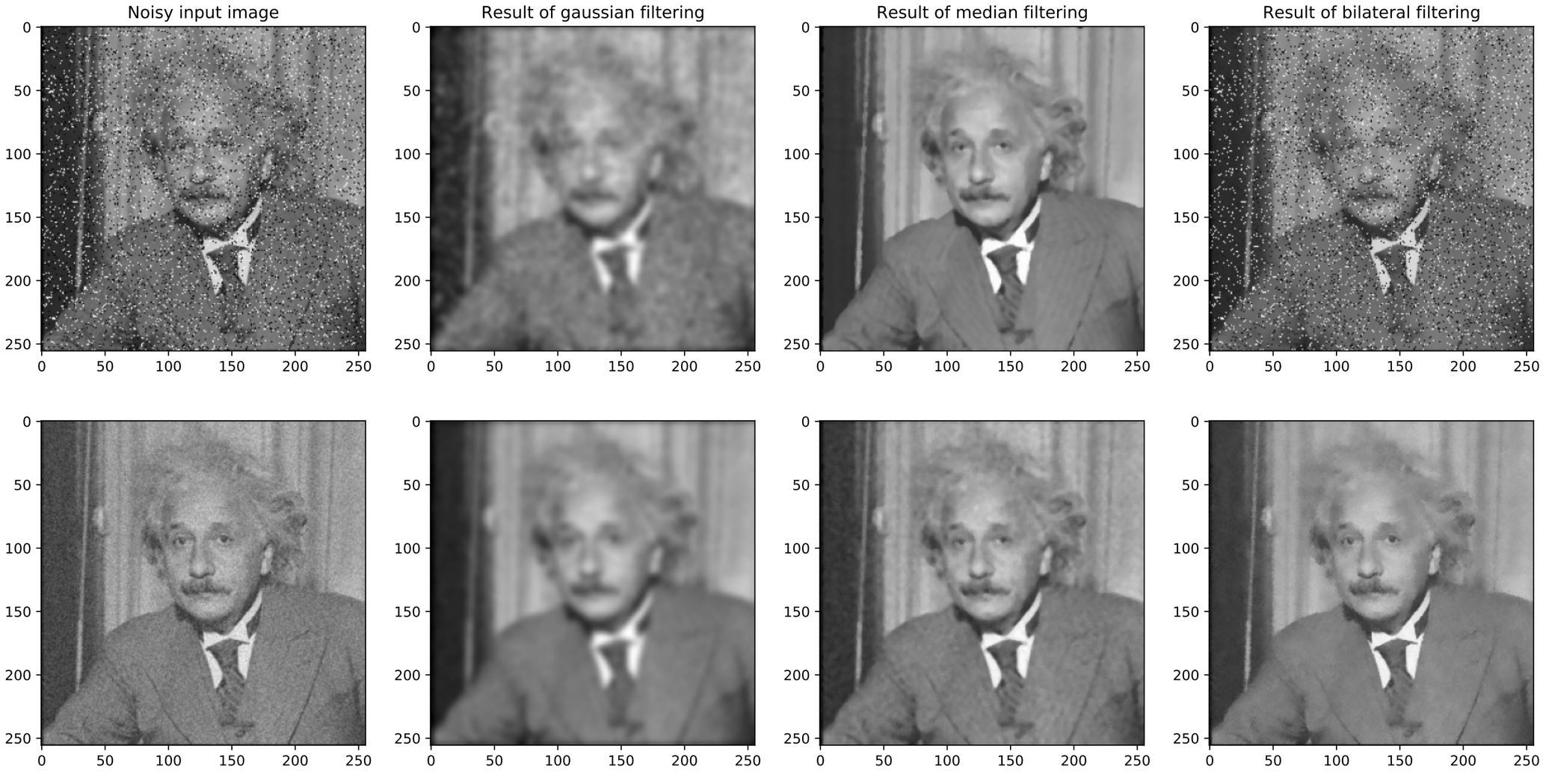
c)

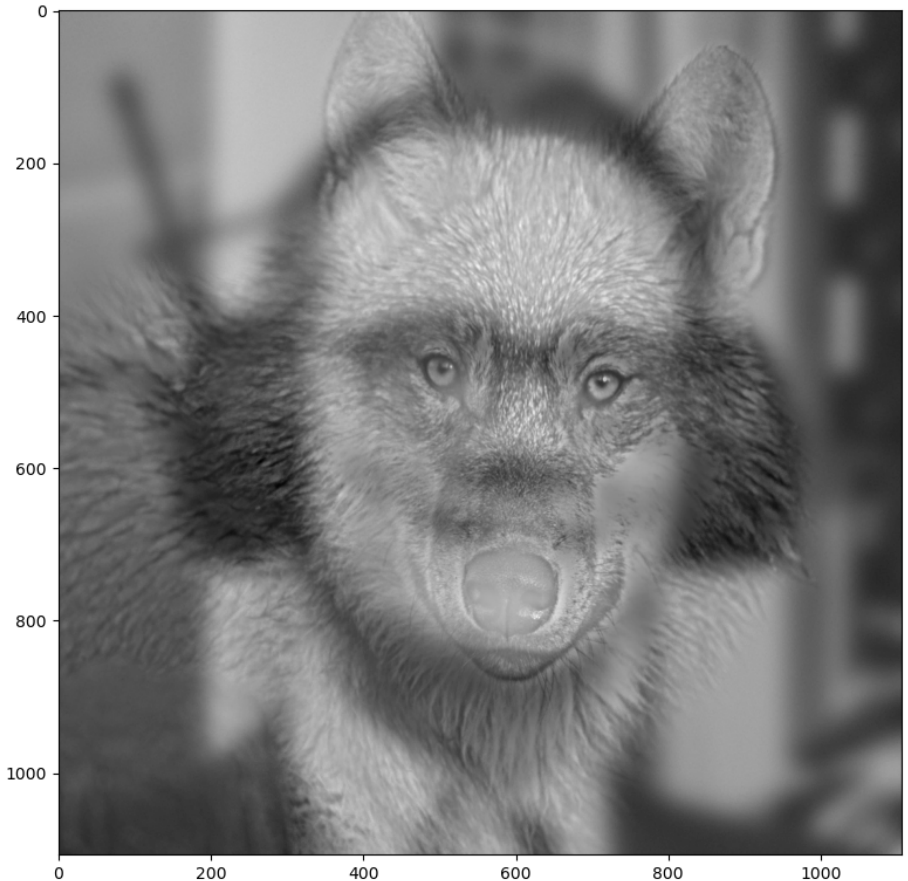
```
p = cross(l1, l2)
```

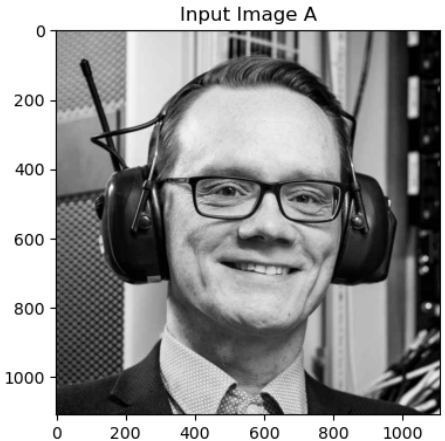
```
p = 3×1
    -7
    -4
    -1
```

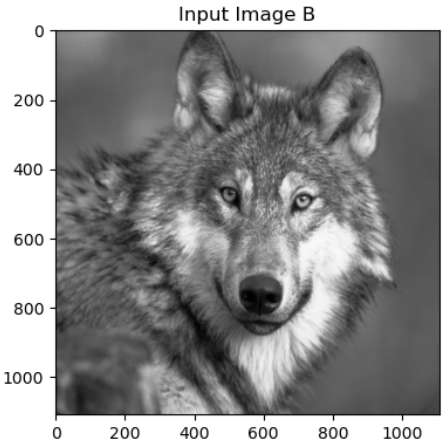
```
[p(1) p(2)]./p(3)
```

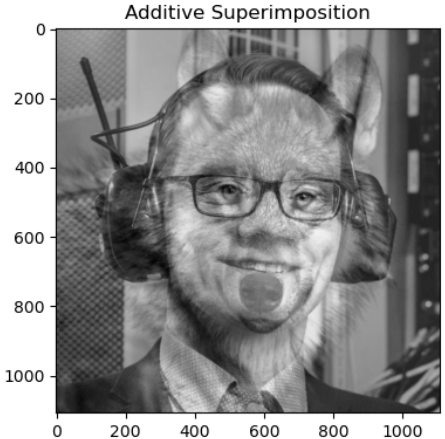
```
ans = 1×2
     7     4
```

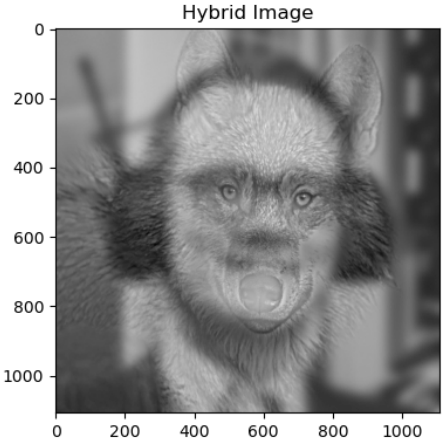

Filtering results
Noisy input image
Result of gaussian filtering
Result of median filtering
Result of bilateral filtering
0
50
100
150
200
250

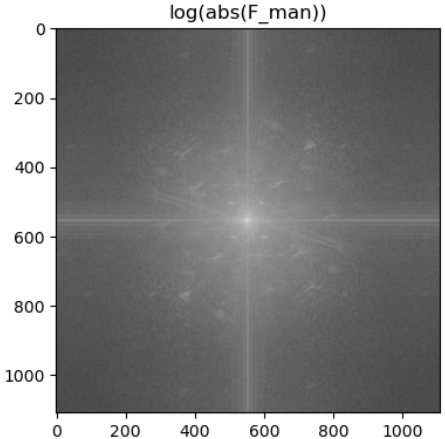

0
200
400
600
800
1000
0
200
400
600
800
1000

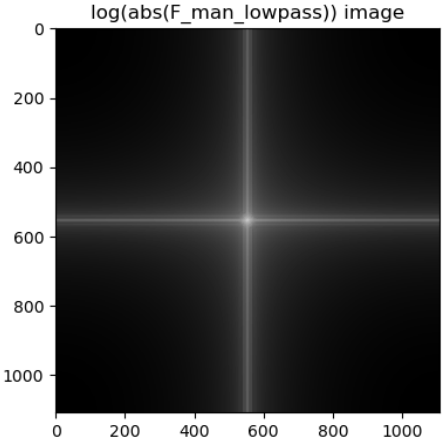

Input Image A
0
200
400
600
800
1000
0
200
400
600
800
1000

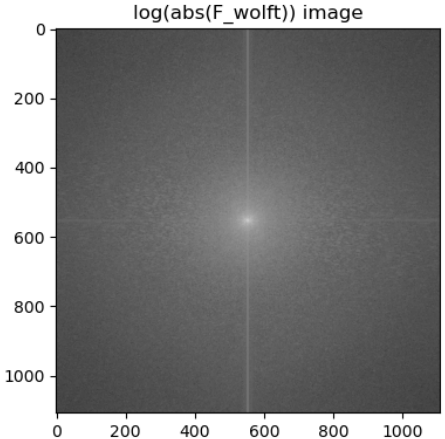

Input Image B
0
200
400
600
800
1000
0
200
400
600
800
1000

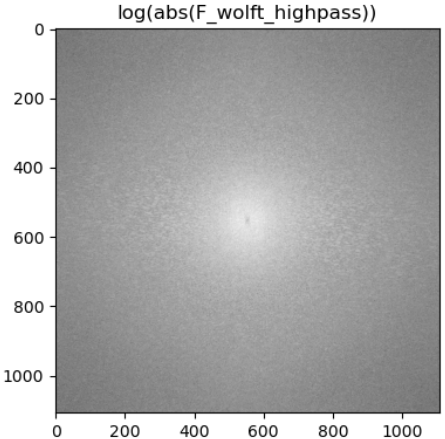

Additive Superimposition
0
200
400
600
800
1000
0
200
400
600
800
1000


Hybrid Image
0
200
400
600
800
1000
0
200
400
600
800
1000


log(abs(F_man))
0
200
400
600
800
1000
0
200
400
600
800
1000


log(abs(F_man_lowpass)) image
0
200
400
600
800
1000
0
200
400
600
800
1000


log(abs(F_wolft)) image
0
200
400
600
800
1000
0
200
400
600
800
1000


log(abs(F_wolft_highpass))
0
200
400
600
800
1000
0
200
400
600
800
1000