Digital Image Processing - Exercise 4

Group C

Ahmad Alabaji
Erik Dietz
Eunbi Park
Timur Pratama Wiradarma

[1] Practice

1) Inverse Filter

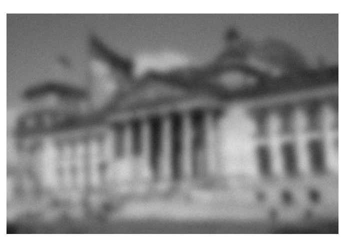
   a. Code (some essential part)

```
//Replace the inverse filter 1/Pi by Qi
for (int ri = 0; ri < freq_kernel.rows; ri++) {
    for (int ci = 0; ci < freq_kernel.cols; ci++) {
        if (norm(freq_kernel.at<Vec2f>(ri, ci)) < threshold) {
            freq_kernel.at<Vec2f>(ri, ci) = threshold;
        }
        std::complex<float> p(freq_kernel.at<Vec2f>(ri,ci)[0], freq_kernel.at<Vec2f>(ri,ci)[1]);
        p = 1.f / p;
                    freq_kernel.at<Vec2f>(ri,ci)[0] = p.real();
                    freq_kernel.at<Vec2f>(ri,ci)[1] = p.imag();
    }
}
```

   - There are some problem of inverse filter : Pi is practically equal to zero in some parts of the spectrum and then Inversion is not feasible due to limited numerical accuracy.
   - In order to solve these problem, replace the inverse filter 1/Pi by Qi

   b. The result(snr =10000)

<original image>



<degraded image>

<inverse filter>

2) Wiener Filter

a. code(some essential part)

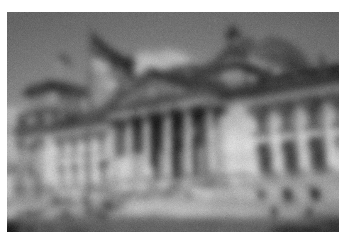
```
 //split kernel into real and imaginary part
 Mat filter_planes[2];
 split(freq_kernel, filter_planes);

//|P|²
 Mat mag_filter;
 magnitude(filter_planes[0], filter_planes[1], mag_filter);
 multiply(mag_filter, mag_filter, mag_filter);
//

//|P|² + 1/SNR²
 Mat denominator = mag_filter + 1 / (snr * snr);
 //Mat denominator = mag_filter;
//

//P*
for (int x = 0; x < freq_kernel.cols; x++) {
 for (int y = 0; y < freq_kernel.rows; y++) {
     //imaginary part
     filter_planes[1].at<float>(y,x) = -1*filter_planes[1].at<float>(y,x);
 }
}

//Q = P* / ( |P|² + 1/SNR² )
  divide(filter_planes[0], denominator, filter_planes[0]);
  divide(filter_planes[1], denominator, filter_planes[1]);
  merge(filter_planes, 2, freq_kernel);
//
```

- Because signal o and noise n are not correlated and no and o are unknown, Q can be formulated by

$$Q_k = \frac{P_k^* |O_k|^2 + N_k^* O_k}{|P_k|^2 |O_k|^2 + |N_k|^2 + P_k O_k N_k^* + P_k^* O_k^* N_k} \Rightarrow \boxed{Q_k = \frac{P_k^*}{|P_k|^2 + 1/SNR^2}}$$

b. The result(snr=10)

<original image>



<degraded image>

<Wiener filter>