EIE4512 - Digital Image Processing

Totorial

Qin Wang

qinwang@cuhk.edu.cn
School of Science and Engineering
The Chinese University of Hong Kong, Shen Zhen

February 28, 2019

Agenda

Corrupt an image with noise

Spatial noise filter

Adaptive spatial filter

Wiener Filter

Bilateral Filter

Corrupt an image with noise

- Degradation Function

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$

- Function Definition

```
g = imnoise(f, type, parameters)
```

- Adding Gaussian noise of mean m, variance var

```
g = imnoise(f, 'gaussian', m, var)
```

- Adding Gaussian noise with zero-mean, local variance V

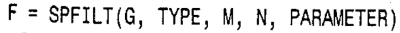
```
g = imnoise(f, 'localvar', V)
```

- Corrupting Image with salt and pepper noise with density d

```
g = imnoise(f, 'salt & pepper', d)
```

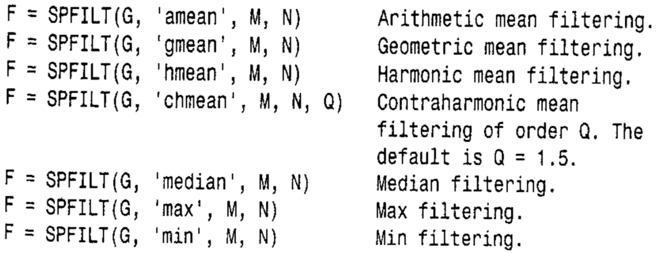
Spatial noise filter

- Spatial filtering on image G using a TYPE filter of size M by N.



```
F = SPFILT(G, TYPE, M, N, PARAMETER)
```

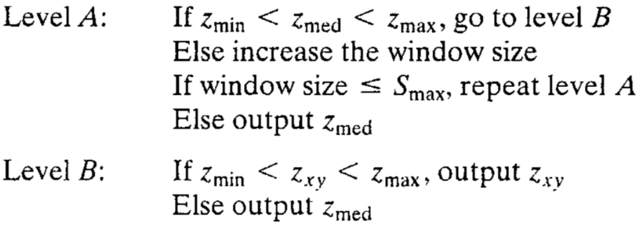
- Ex:following



```
F = SPFILT(G, 'amean', M, N)        Arithmetic mean filtering.
F = SPFILT(G, 'gmean', M, N)        Geometric mean filtering.
F = SPFILT(G, 'hmean', M, N)        Harmonic mean filtering.
F = SPFILT(G, 'chmean', M, N, Q)    Contraharmonic mean
                                    filtering of order Q. The
                                    default is Q = 1.5.
F = SPFILT(G, 'median', M, N)       Median filtering.
F = SPFILT(G, 'max', M, N)          Max filtering.
F = SPFILT(G, 'min', M, N)          Min filtering.
```

Adaptive Spatial Filter

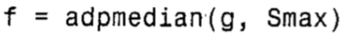
- Algorithm



```
Level A:    If $z_{\min} < z_{\text{med}} < z_{\max}$, go to level $B$
            Else increase the window size
            If window size $\leq S_{\max}$, repeat level $A$
            Else output $z_{\text{med}}$

Level B:    If $z_{\min} < z_{xy} < z_{\max}$, output $z_{xy}$
            Else output $z_{\text{med}}$
```

- An M-function implements Adaptive Spatial Filter, g is the input image, Smax is the maximum size of the adaptive filter window.



```
f = adpmedian(g, Smax)
```

Wiener Filter

- Formula

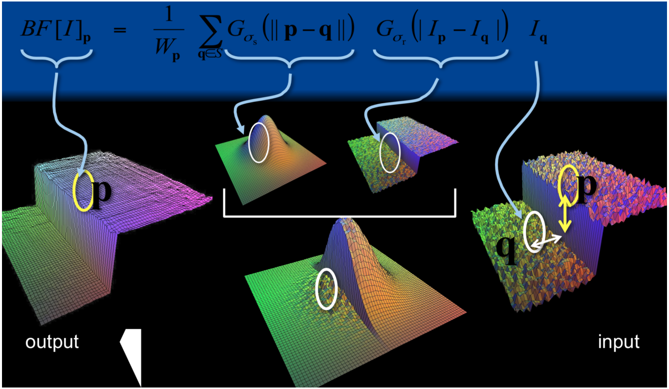
$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

- Wiener filtering is implemented in IPT using function deconvwnr

```
J = deconvwnr(I,PSF)
J = deconvwnr(I,PSF,NSR)
J = deconvwnr(I,PSF,NCORR,ICORR)
```

Bilateral Filter

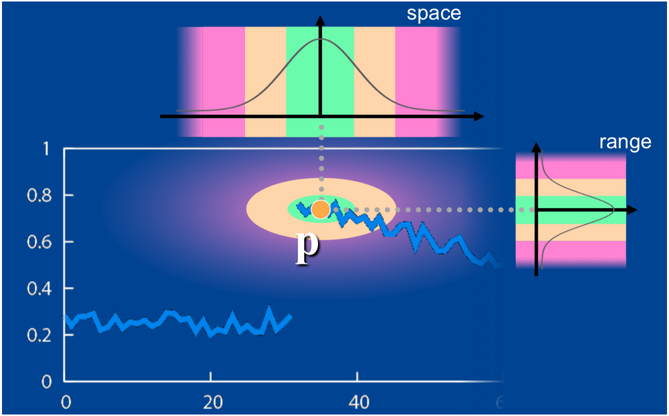
- Bilateral filtering of images with Gaussian kernels.



$$\underbrace{BF[I]_{\mathbf{p}}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in S} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)} \; \underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)} \; I_{\mathbf{q}}$$

Bilateral Filter

- Sigma Parameters turning.



Practice

- Trying to implement Bilateral Filter by yourself.(note: reference the code on the GitHub)