Tutorial 5 - Image Restoration, Filtering and diffusion

COMP 4421: Image Processing

October 8, 2019

Previous

- Noise Models
- Restoration in the Presence of Noise Only-Spatial Filtering

Outline

- Restoration based on Degradation Function
- Periodic Noise Reduction by Frequency Domain Filtering
    - Bandreject Filters
    - Bandpass Filters
    - Notch Filters
- Linear Diffusion

Restoration based on Degradation Function

- Inverse filtering
- Minimum mean square error (Wiener) filtering
- Geometric mean filter

Degradation Model

- Spatial domain
  $g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$
- Frequency domain
  $G(x,y) = H(x,y)F(x,y) + N(x,y)$
- Problem Definition
  - Observe the degraded image $g(x,y)$
  - H function and N (noise) are either known or need to be estimated
  - Goal: restore the original image $f(x,y)$

Matlab built-in functions

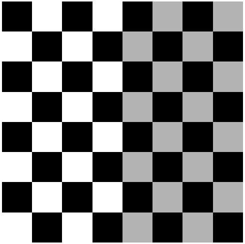
- PSF = fspecial('type', paras);
  - All kinds of spatial filters.
    Type includes 'average', 'disk', 'gaussian', 'motion' $\cdots$
  - Paras are corresponding parameters
  - PSF is the spatial domain filter created
  - Type help fspecial for more
- fr = deconvwnr(g, PSF, NSPR);
  - Wiener filtering: g is corrupted image, fr is restored one
  - PSF is the filter that degrades the image
  - NSPR: noise to signal power ratio
  - Type help deconvwnr for more

Inverse Filtering

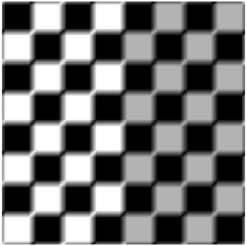
Matlab Code

```
f = checkerboard(20);
figure,subplot(1,3,1); imshow(f, [ ]); xlabel('original image')
PSF = fspecial('motion',7,45);
gb = imfilter(f,PSF,'circular');
subplot(1,3,2); imshow(gb, [ ]); xlabel('motion blurred')
fr = deconvwnr(gb,PSF);
subplot(1,3,3); imshow(fr, [ ]); xlabel('restored');
```

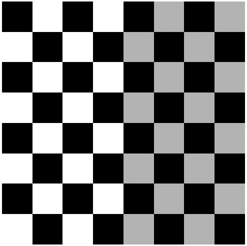
Inverse Filtering



original image



motion blurred



restored

Wiener Filtering

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

- $S_n(u,v)$: power spectrum of the noise
- $S_f(u,v)$: power spectrum of the true, un-degraded, uncorrupted image

Wiener Filtering

- If neither N nor F is known, then the solution is to approximate

$$\frac{S_\eta(u,v)}{S_f(u,v)} = K$$

- The approximated, estimated image is given by

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

Wiener Filtering

```
f = checkerboard(8);
PSF = fspecial('motion',7,45);
gb = imfilter(f,PSF,'circular');
noise = normrnd(0,0.001^0.5,size(f));
gb = gb + noise;
figure,subplot(1,3,1); imshow(gb, [ ]); xlabel('noisy image');
fr1 = deconvwnr(gb, PSF);
subplot(1,3,2),imshow(fr1, [ ]); xlabel('inverse filtering');
Sn = abs(fft2(noise)).^2; %noise power spectrum
nA = mean(Sn(:)); % noise mean power spectrum;
Sf = abs(fft2(f)).^2; %image power spectrum
fA = mean(Sf(:)); %image mean power spectrum
K = nA/fA;
fr2 = deconvwnr(gb,PSF,K);
subplot(1,3,3); imshow(fr2,[]); xlabel('Wiener filtering');
```

Periodic Noise Reduction by Frequency Domain Filtering

- Periodic Noise Reduction by Frequency Domain Filtering
  - Bandreject Filters
  - Bandpass Filters
  - Notch Filters

Periodic Noise Reduction by Frequency Domain Filtering

- Periodic Noise Reduction by Frequency Domain Filtering
  - Bandreject Filters
  - Bandpass Filters
  - Notch Filters

Bandreject Filters

Bandreject filters

- Bandreject filters remove a band of frequencies about the origin of the frequency representation (or Fourier spectrum).
- Ideal Bandreject Filter

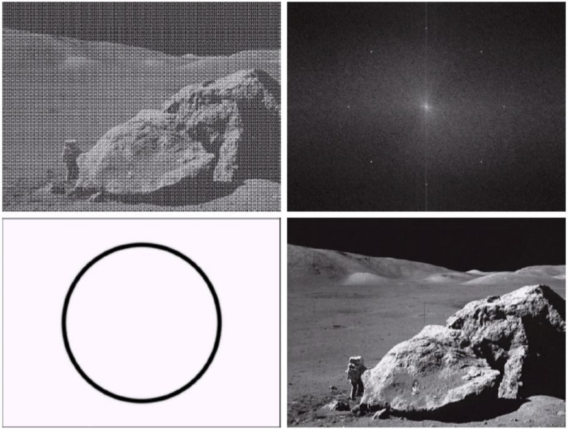
$$H(u,v) = \begin{cases} 1 & \text{if} \quad D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if} \quad D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if} \quad D_0 + \frac{W}{2} < D(u,v) \end{cases}$$

W = width of the band
D_0 = radial centre

Bandreject Filters

Butterworth Bandreject Filter

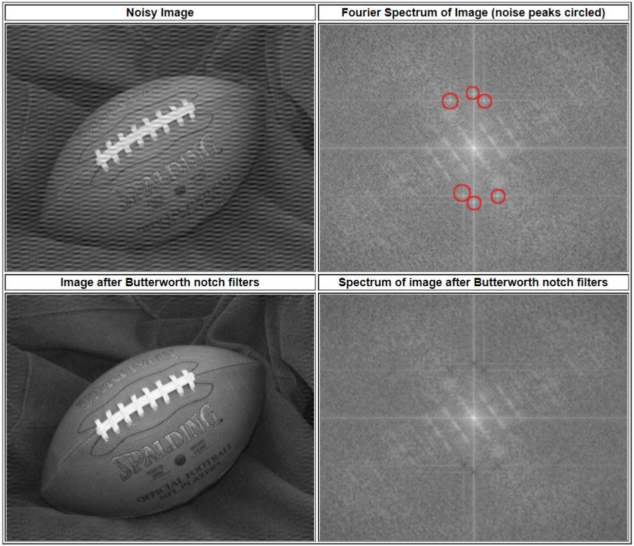
- It is a sharp, narrow filter in frequency domain.
- order n = 4



a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Bandreject Filters



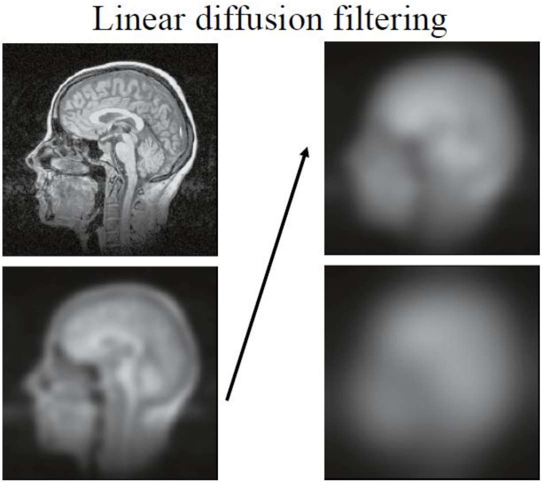
Diffusion

- First law: mass flux (diffusion) is proportional to the concentration gradient (change in concentration)
  - $\vec{\mathcal{J}} = -D \cdot \ \nabla u$ and $\nabla u = \frac{\partial u}{\partial x}$
- Second law: the rate of accumulation of concentration within a volume is proportional to the change of local concentration gradient (continuity equation).
  - $\frac{\partial u}{\partial t} = -\frac{\partial \mathcal{J}_x}{\partial x} = -\frac{\partial}{\partial x}(-D \cdot \nabla u) = D\frac{\partial^2 u}{\partial x^2}$

Linear Diffusion

- Linear diffusion filtering: $D = I$ (Identity matrix), initial condition $u(\vec{x}, t = 0) = I(\vec{x})$
- u in the next time step:
  $u(x, t + \Delta t) = u(x, t) + \frac{\Delta t}{(\Delta x)^2}(u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t))$

Linear Diffusion



Linear diffusion examples: t=0, 12.5, 50, 200.

Thank you!