Lab 9. (case) 3D deconvolution of the microscopy fluorescence image

Materials

- Five ‘*.mat’ files
  - simSphere1.mat - a non-blurred microsphere
  - PSFG&Lsmall.mat - a calculated PSF data
  - simblurNoNoise.mat - a calculated blurred microsphere without noise

  - Realdat.mat - a measured microsphere data
  - PSFreal.mat - a measured PSF data

Tasks

1. Generate noise disturbed data using calculated blurred microsphere data;
   - Be careful, the original calculated blurred microsphere data assumes that the fluorescence intensity of each point is 1 and this fluorescence intensity should be adjusted to ensure the final value of image pixels are not exceed 1.
   - It is recommended to add background noise to simulate the real detector system.
2. Implement the inverse filtering and Wiener filtering method;
   - MATLAB function ‘deconvwnr’ is not allowed.
3. Implement the R-L iteration algorithm. (Optional)

Tasks(cont.)

4. Compare the inverse filtering, Wiener filtering and R-L iteration on simulated data;
    - the R-L iteration method can applied based on either of TASK 3 and MATLAB function ‘deconvlucy’;
    - try to change the noise model and SNR to compare and discuss the different results.
5. Apply the inverse filtering, Wiener filtering and R-L iteration on real data and compare the results.

Methods and Tips

Inverse filtering

$$\bar{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

$$\bar{f}(x,y,z) = \mathcal{F}^{-1}\left\{\frac{G\left(\omega_x,\omega_y,\omega_z\right)}{H\left(\omega_x,\omega_y,\omega_z\right)}\right\}$$

Weiner filtering

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

The inverse filter in Weiner filtering

$$\hat{H}(\omega_x, \omega_y, \omega_z) = \frac{H^*(\omega_x, \omega_y, \omega_z)}{\left(\left|H(\omega_x, \omega_y, \omega_z)\right|^2 + \boxed{\frac{S_\eta(\omega_x, \omega_y, \omega_z)}{S_f(\omega_x, \omega_y, \omega_z)}}\right)}$$

NSR

Tips

- $S_f(\omega_x, \omega_y, \omega_z)$
  - Power of the non-blurred signal

$$\frac{S_\eta(\omega_x, \omega_y, \omega_z)}{S_f(\omega_x, \omega_y, \omega_z)}$$

- $S_\eta(\omega_x, \omega_y, \omega_z)$
  - Power of the noise

- When NSR = 0
  - Inverse filtering = Weiner filtering

Tips

- For simulated / calculated data
  - Be careful, the original calculated blurred microsphere data assumes that the fluorescence intensity of each point is 1 and this fluorescence intensity should be adjusted to ensure the final value of image pixels does not exceed 1.
  - It is recommended to add background noise to simulate the real detector system.
  - blurred * factor + background noise < 1
  - Power of (non-blurred signal*factor)
- For real data
  - Power of non-blurred signal?

Tips

$$\hat{H}(\omega_x, \omega_y, \omega_z) = \frac{H^*(\omega_x, \omega_y, \omega_z)}{\left(\left|H(\omega_x, \omega_y, \omega_z)\right|^2 + \frac{S_\eta(\omega_x, \omega_y, \omega_z)}{S_f(\omega_x, \omega_y, \omega_z)}\right)}$$

- Avoiding extremely small numbers in the denominator
  - Set these numbers to be numbers a little bigger
  - Useful Matlab functions: eps, realmin, …
    - eps = 2.2204e-16
    - sqrt(eps) = 1.4901e-08
    - realmin = 2.2251e-308

Maximize the likelihood

- Maximize the log likelihood of the Poisson process

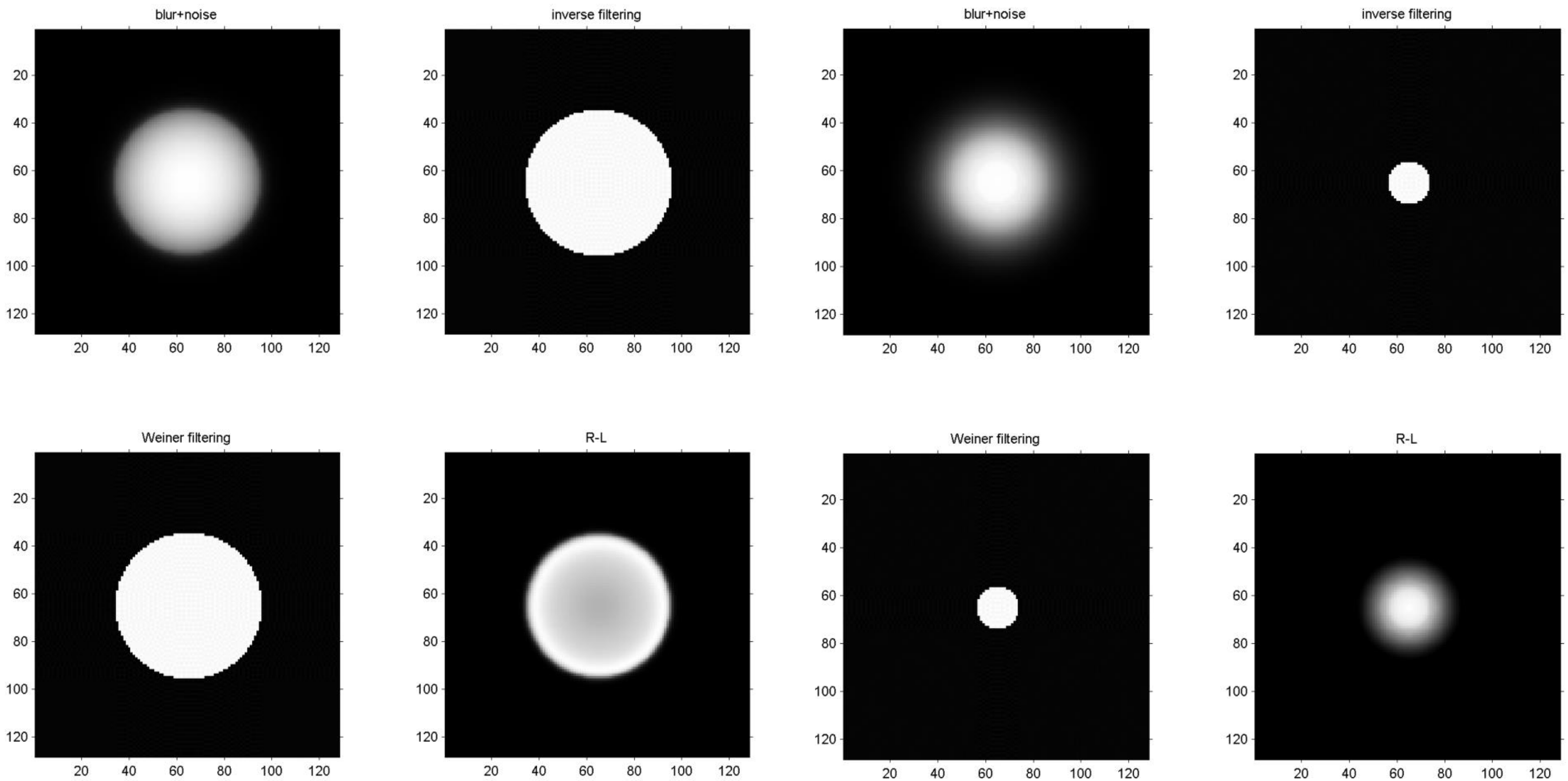
$$\max J(f) = \sum_{x,y,z} g * log\lambda - \lambda$$

- Apply the gradient-based iterative search algorithm(R-L iteration algorithm)

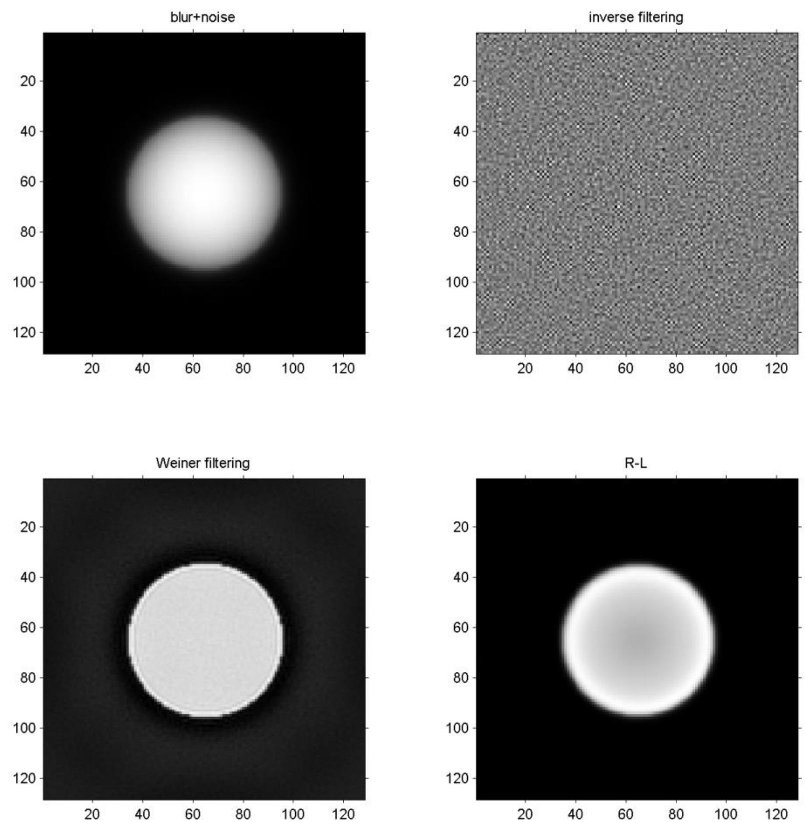
$$\bar{f}_{k+1} = \left[\frac{g}{\bar{f}_k \otimes h} \otimes (-h)\right] \bar{f}_k, \qquad (-h) = h(-x, -y, -z)$$

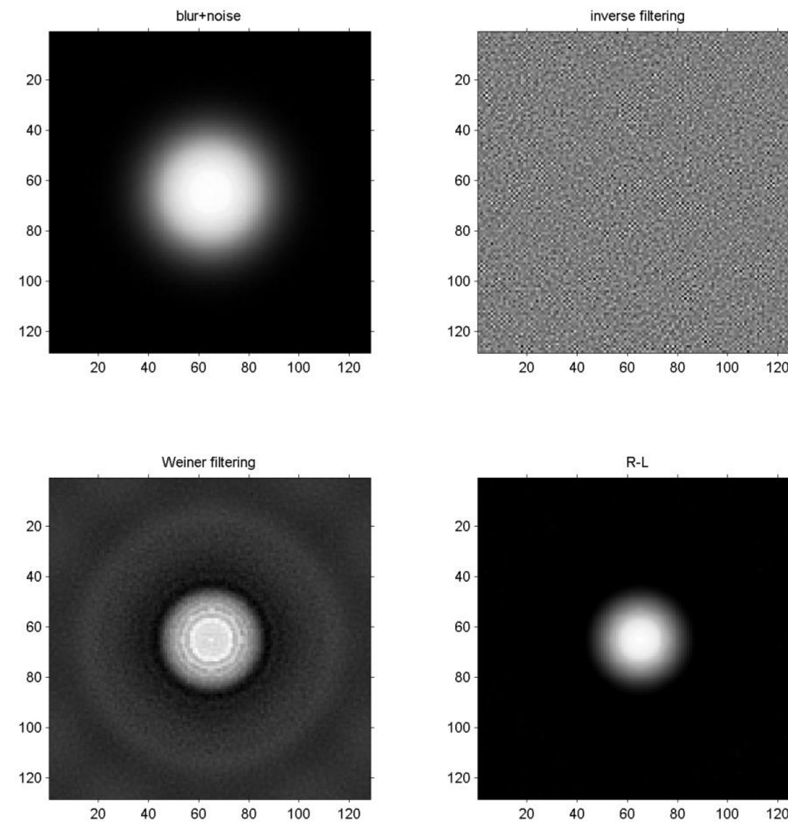
- imnoise
  - Add noise to image
- psf2otf
  - Convert point-spread function to optical transfer function
- fftn, ifftn
  - n-D fft, ifft
- flipdim
- deconvlucy

Examples

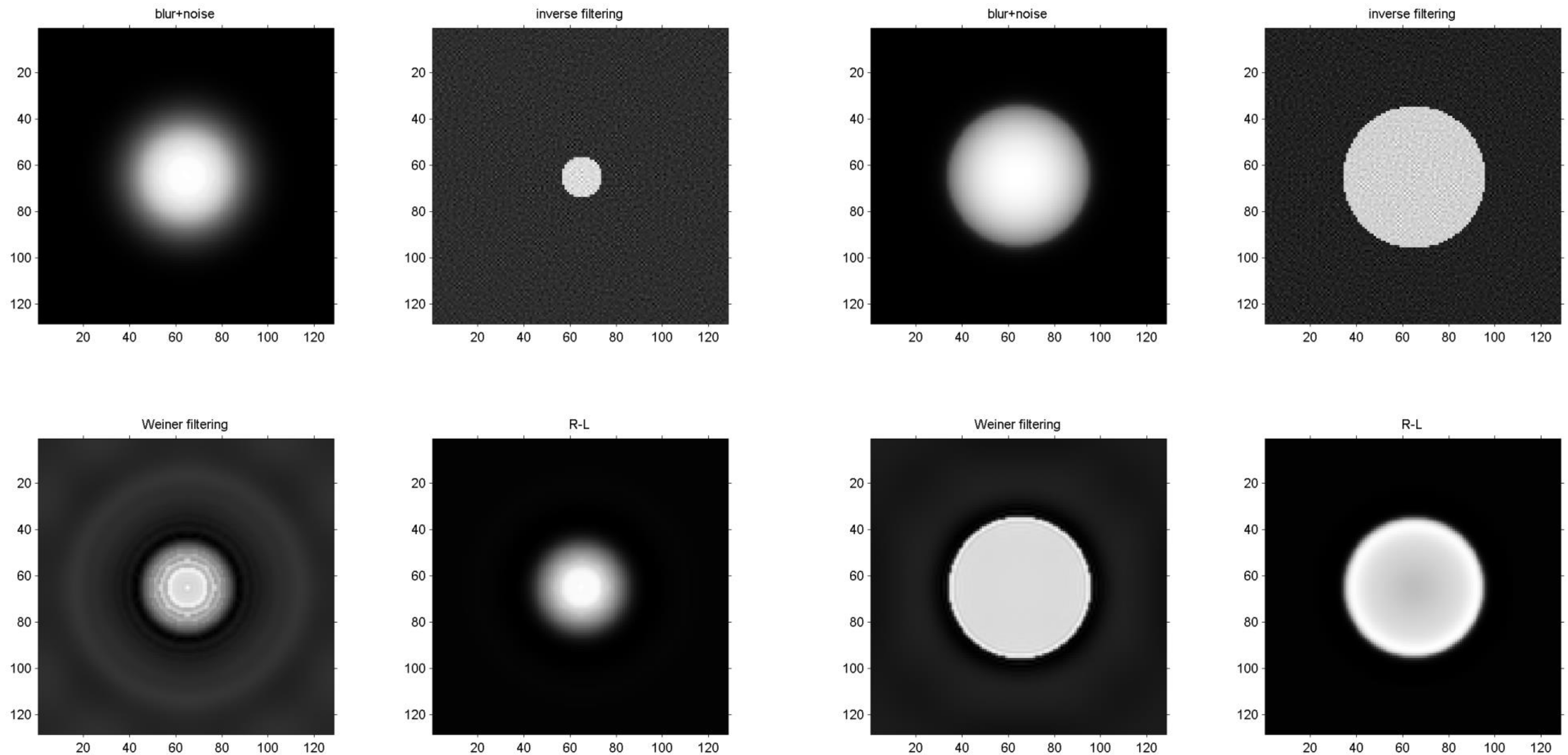

blur+noise
inverse filtering
blur+noise
inverse filtering
Weiner filtering
R-L
Weiner filtering
R-L

Examples


blur+noise
inverse filtering
Weiner filtering
R-L


blur+noise
inverse filtering
Weiner filtering
R-L

Examples


blur+noise
inverse filtering
blur+noise
inverse filtering
Weiner filtering
R-L
Weiner filtering
R-L

The end