Deconvolutions using Wiener filter and Richardson-Lucy algorithm

Sushko Nikita

June 26, 2022

Table of Contents

Entropy

The formula for entropy of a signal is:

$$H = -\sum_{k} p_k \log p_k,$$

where k is the number of grey levels in the image and p_k is the probability associated with grey level k.

Total Variation

Total Variation is a regularisation technique that imposes spatial continuty of the neighboring pixels, penalizing the outliers. The formula for the 2-D Total Variation is:

$$V(X) = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2},$$

where x is a 2-D signal, i.e. an image.
Since substraction of the neighboring pixels of the x and y axis is basically Sobel filtering, i.e. computing the gradients of the image, we can rewrite the formula as:

$$V(X) = \sum_{i,j} \sqrt{|\nabla_i x_{i,j}|^2 + |\nabla_j x_{i,j}|^2}.$$

Total Variation

Sometimes, an easier to minimize variation of the formula is used:

$$V(X) = \sum_{i,j} \sqrt{|\nabla_i x_{i,j}|^2} + \sqrt{|\nabla_j x_{i,j}|^2} = \sum_{i,j} |\nabla_i x_{i,j}| + |\nabla_j x_{i,j}|.$$

Total Variation denoising

Total Variation can also be used as a denoiser. By formulating the optimization problem as:

$$\min_{y} ||x - y|| + \lambda V(y),$$

where $||x - y||$ is the 2-D L_2-norm, λ is a parameter and $V(y)$ is the Total Variation and y is the noisy image, we can denoise the image while maintaining the edge information.

Total Variation denoising

Figure: The result of Total Variation denoising

Richardson-Lucy

The data model for the 1-D signal is:

$$d_i = \sum_j p_{i,j} u_j,$$

where u_j is the intensity of the signal at the point j, d_i is the detected intensity at the point i and $p_{i,j}$ is a coefficient from the transition matrix P that describes the portion of signal from the source j that is detected in point i.

Richardson-Lucy

In order to estimate u_j if the PSF is known and we have the observed signal d_i, we can begin iterative process that is described as follows:

$$\hat{u}_j^{(t+1)} = \hat{u}_j^{(t)} \sum_i \frac{d_i}{\sum_j p_{ij} \hat{u}_j^{(t)}} p_{ij},$$

where t is the number of the iteration.
We can rewrite the equation in terms of convolution with a PSF for the 2-D signal the following way:

$$\hat{u}^{(t+1)} = \hat{u}^{(t)} \left(\frac{d}{\hat{u}^{(t)} \otimes P} \otimes P^T \right),$$

where $\otimes$ is a 2-D convolution of an image, P^T is the transposed PSF and the division and multiplication are element wise.

Richardson-Lucy

We can introduce the Total Variation regularization to the process, if we rewrite the iterative equation as:

$$\hat{u}^{(t+1)} = \frac{\hat{u}^{(t)}}{1 - \lambda V}\left(\frac{d}{\hat{u}^{(t)} \otimes P} \otimes P^T\right).$$

Wiener

The data model for the signal is:

$$y = hx + n,$$

where y is the resulting signal, h is the Point Spread Function, x is the unknown original signal and n is the noise.

Wiener

The Wiener filter in the frequency domain can be defined as

$$G(f) = \frac{H^*(f)S(f)}{|H(f)|^2 S(f) + N(f)},$$

where:

- $G(f)$ and $H(f)$ are the Fourier transforms of the Wiener deconvolution filter and the PSF,
- $S(f) = \mathbb{E}|X(f)|^2$ is the mean power spectral density of the original signal x,
- $N(f) = \mathbb{E}|X(f)|^2$ is the mean power spectral density of the noise n,
- $H^*(f)$ is the complex conjugation of $H(f)$.

Wiener

Thus, we can carry out the operation in the frequency domain as following:

$$\hat{X(f)} = G(f)\,Y(f),$$

and then perform an inverse Fourier transform, to obtain x.

Data

For the data, The Mouse hematopoietic stem cells in hydrogel microwells dataset from the Cell Tracking Competition was used. The images are greyscale, 1010x1010 pixels in size.

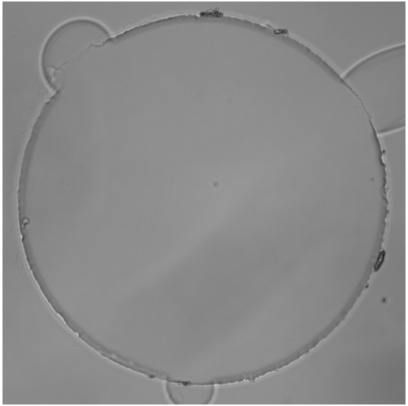


Figure: An example from the dataset, entropy is 4.8303

Preprocessing

Before exploring deconvolution methods, some preprocessing was done. According to the formulas, we have to apply some PSF to the images and then add some noise.
A gaussian blur kernel was used as the PSF function for the images. Here is the result of blurring the image:

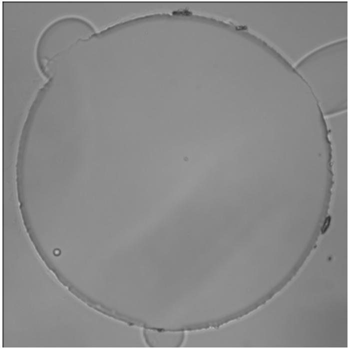
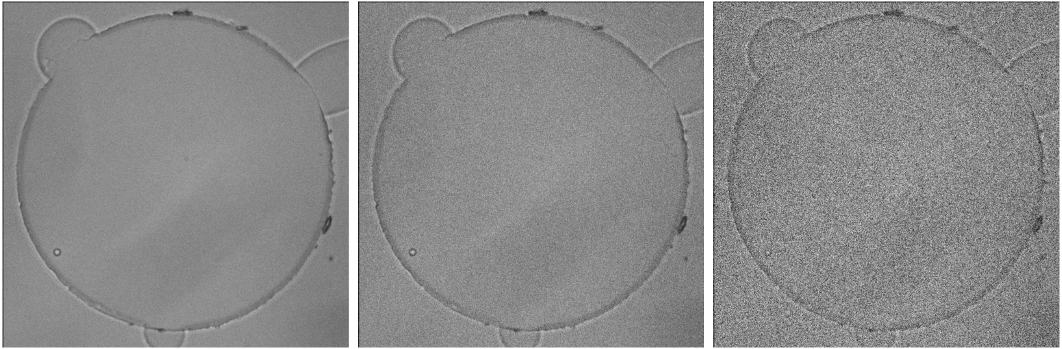


Figure: Filter size is 5, sigma is 10, entropy is 19.4360

Preprocessing



(a) Std = 0.01, entropy = 19.9601

(b) Std = 0.11, entropy = 19.9565

(c) Std = 0.3, entropy = 18.3776

Figure: Images with different parameters of gaussian noise

Preprocessing

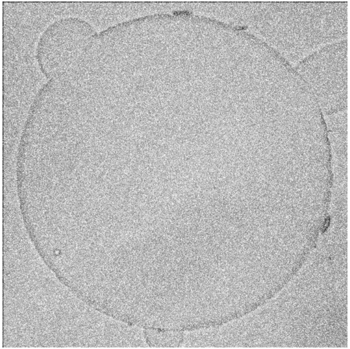
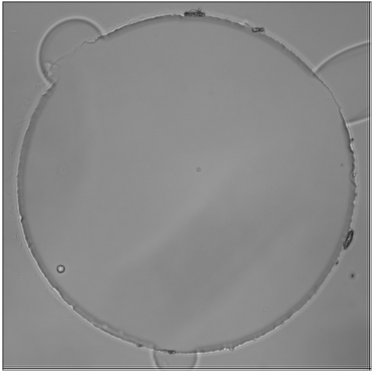
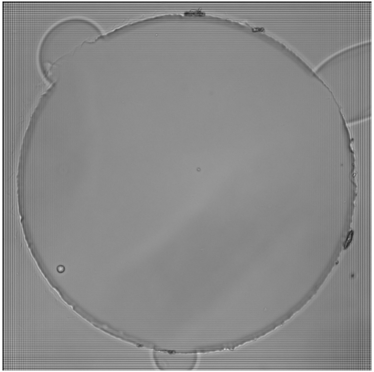


Figure: An image with added poisson noise, entropy is 12.0254

Restoration



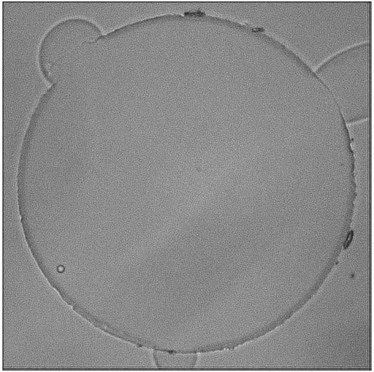
(a) Unblurred using Richardson-Lucy, SSIM = 0.9463, PSNR = 20.0621



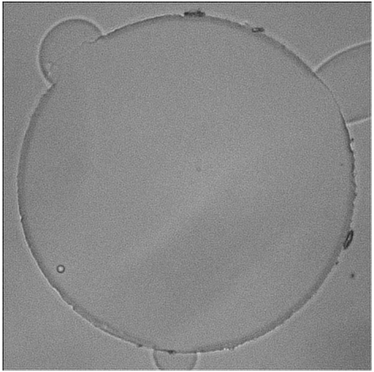
(b) Unblurred using Wiener, SSIM = 0.6798, PSNR = 18.0324

Figure: Deconvolutions of images without any noise

Restoration



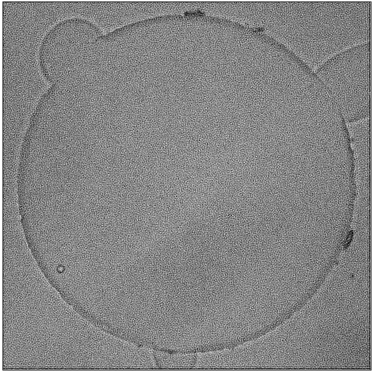
(a) Unblurred using Richardson-Lucy, SSIM = 0.1808, PSNR = 16.7633



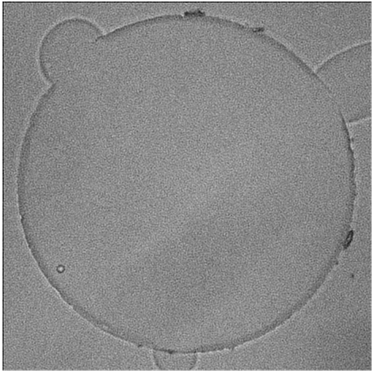
(b) Unblurred using Wiener, SSIM = 0.7064, PSNR = 25.3072

Figure: Deconvolutions of images with gaussian noise with sigma = 0.01

Restoration



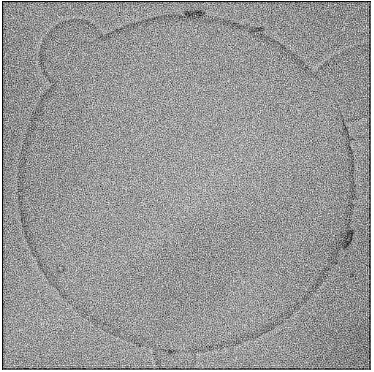
(a) Unblurred using Richardson-Lucy, SSIM = 0.0583, PSNR = 11.9206



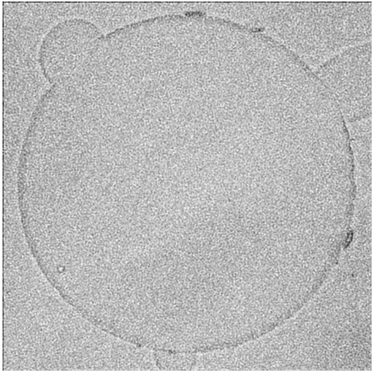
(b) Unblurred using Wiener, SSIM = 0.5456, PSNR = 23.3830

Figure: Deconvolutions of images with gaussian noise with sigma = 0.11

Restoration



(a) Unblurred using Richardson-Lucy, SSIM = 0.0355, PSNR = 9.7349



(b) Unblurred using Wiener, SSIM = 0.3611, PSNR = 13.4614

Figure: Deconvolutions of images with Poisson noise

Conclusions

- Richardson-Lucy method of deconvolution has no edge effects on non-noisy images, opposed to Wiener filtering,
- Both PSNR and SSIM metrics are far better with Wiener filtering,
- Both Wiener filtering and Richardson-Lucy method are vulnerable to noise.

(a) Github page with the code

(b) HTML page for quick view of the notebook