Acknowledgement

Figures from Gonzales and Woods, Digital Image Processing, 3E, Prentice Hall

Outline

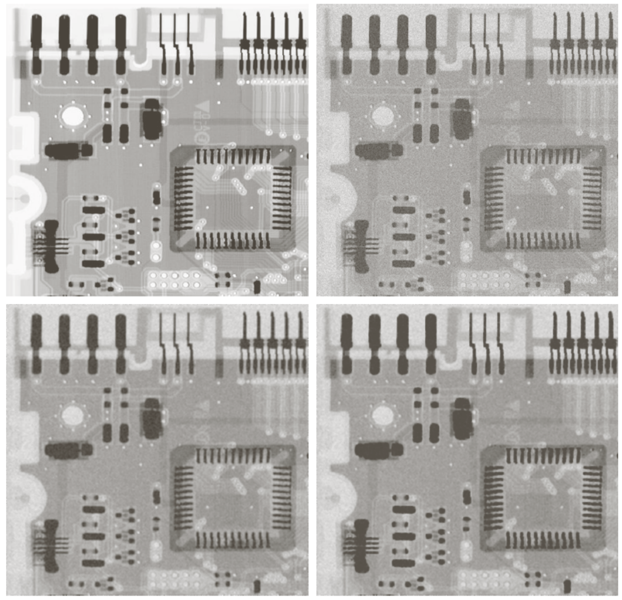
Contents

1 Introduction

Image Enhancement and Restoration

- Enhancement: Improve the overall image quality for (mostly) human or computer processing. Input is an image and the output is a ‘visually better quality” image.
- Broadly categorized as **pixel based** or **region based** operations.
- Image Restoration: *knowledge-based enhancement.* Here you use prior knowledge about how the image is degraded in order to improve the visual quality. Restoration is also an image-to-image transformation, with the output being a better quality image.

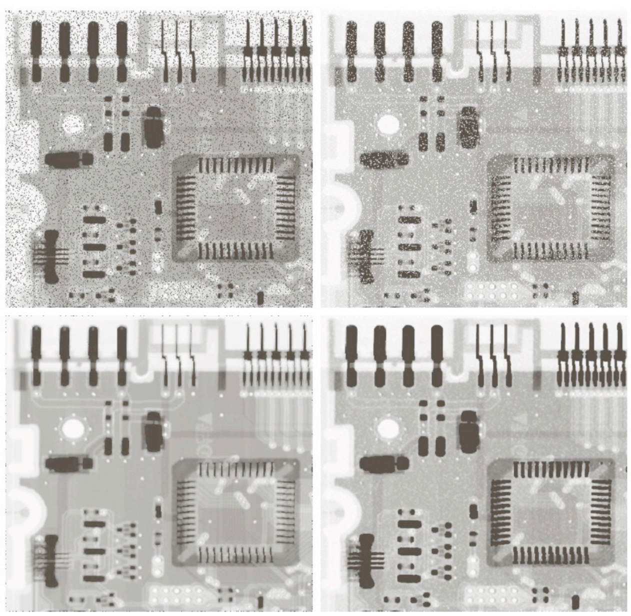
Examples: Additive Gaussian Noise



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

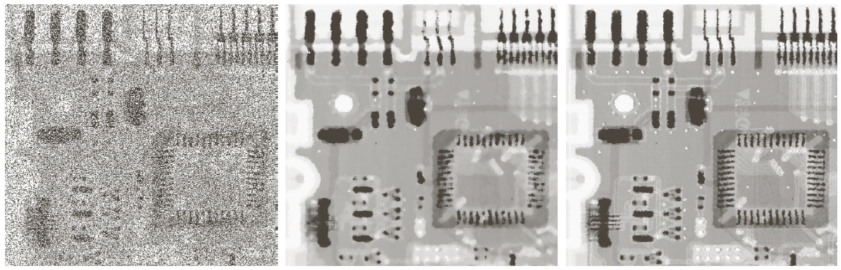
Examples: Salt & Pepper Noise



a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contra-harmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Examples: Adaptive median filter



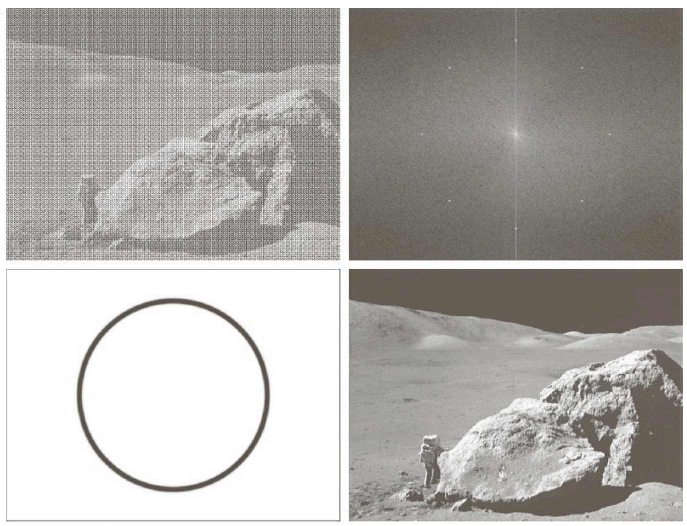
a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

Examples: Sinusoidal Noise and Band-reject filter

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.)



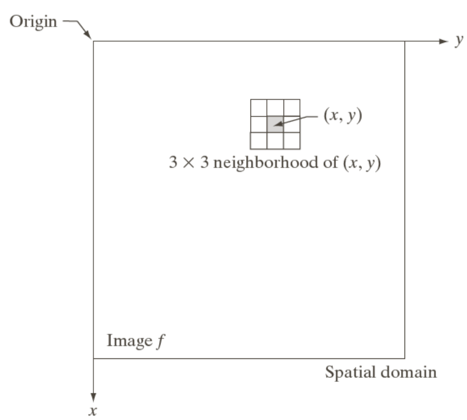
Examples: Motion Blur

Examples: deblurring magic! Inverse and Wiener filtering

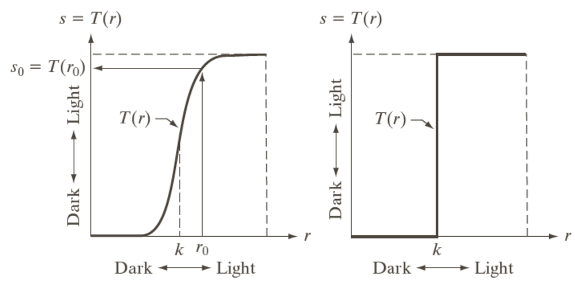
2 Image Enhancement

2.1 Pixel-level operations

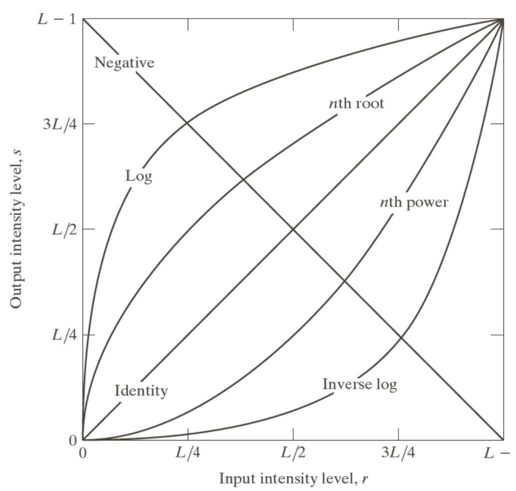
Pixel Neighborhood



Contrast stretching and Thresholding



Intensity transformations



Negative Picture: $s = L - 1 - r$

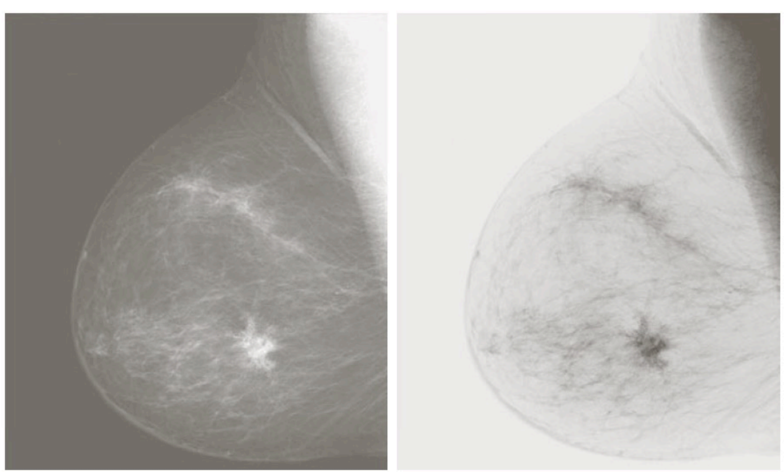
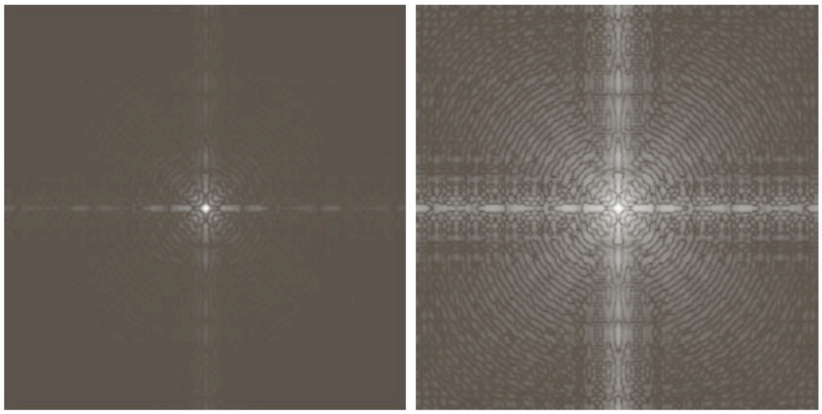


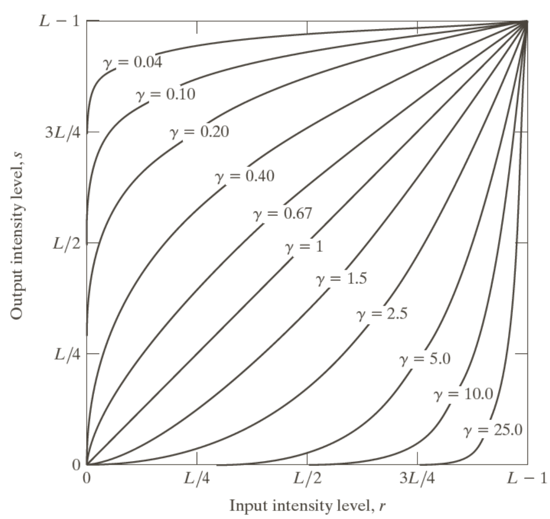
Figure shows a mammogram and its negative

Log scaling: $s = c\log(1 + r)$



Enhanced Fourier transform using log scaling, $c = 1$.

Gamma transformation: $s = cr^{\gamma}$

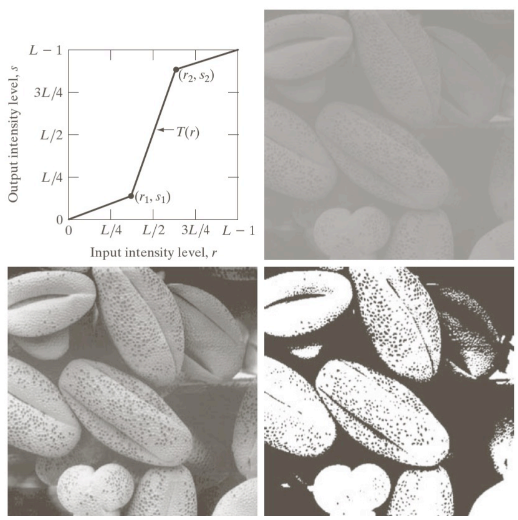


Gamma stretching

15. Restoration and Wiener Filtering

Top left: Original image; $c = 1, \gamma = 3, 4, 5$

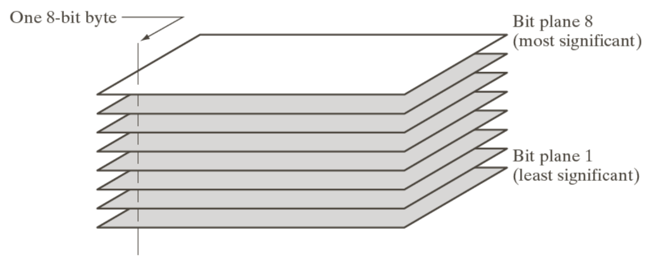
Contrast stretching



Bottom left: after stretching using (a); Bottom right: thresholding

Bit-plane Slicing

15. Restoration and Wiener Filtering



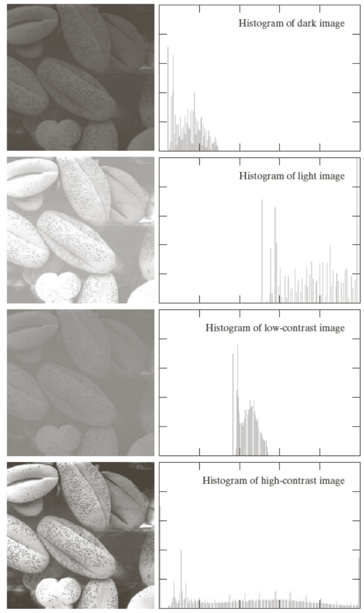
Bit-plane Slicing: example

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

2.2 Histogram Processing Operations: Equalization

Different contrast images



Histogram Equalization

Let r denote the image intensities to be transformed. We will assume that $r \in [0, L-1]$, with $r = 0$ representing black and $r = L-1$ representing white. These are typically 0 and 255, respectively.

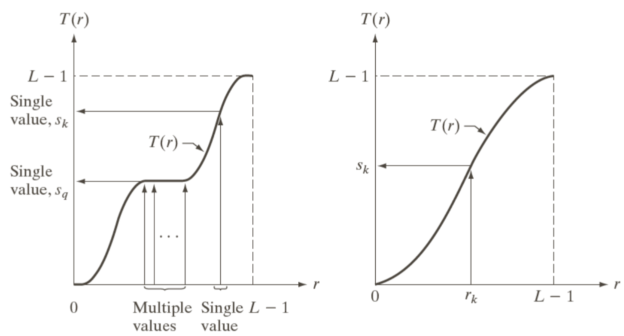
We want to transform the intensities r to a new set of values s according to

$$s = T(r), 0 \leq r \leq L-1 \tag{1}$$

s represent the transformed (output) image intensities. Further, we assume that

1. $T(r)$ is strictly monotonically increasing function of r in the interval $[0, L-1]$, and
2. $0 \leq T(r) \leq L-1$ for $0 \leq r \leq L-1$
3. and that $r = T^{-1}(s), 0 \leq s \leq L-1$

$s = T(r)$



Histograms as probability density functions (PDF)

Let $p_r(r)$ and $p_s(s)$ represent the PDFs of the random variables r and s, respectively.

Given $p_r(r)$, you can calculate $p_s(s)$ as

$$p_s(s) = p_r(r)|\frac{dr}{ds}| \tag{2}$$

Now consider (the CDF: Cumulative Distribution Function)

$$s = T(r) = \int_0^r p_r(w)dw \tag{3}$$

Histogram Equalization vis CDF

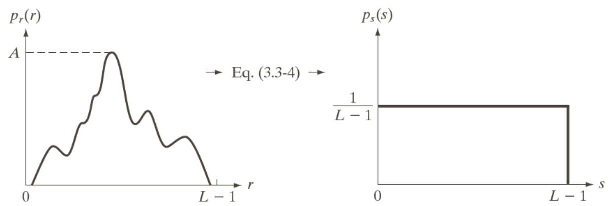
For the CDF,

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = \frac{d}{dr}[\int_0^r p_r(w)dw] = p_r(r) \tag{4}$$

Thus,

$$p_s(s) = p_r(r)|\frac{dr}{ds}| = 1 \tag{5}$$

The transformation corresponding to CDF equalizes the histogram! Note that the range in [0, 1], so you can scale this to the original image pixel intensity range by multiplying by, say, 255.



a b

FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, *r*. The resulting intensities, *s*, have a uniform PDF, independently of the form of the PDF of the *r*'s.

Example: 64 x 64 image, L=8

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

You can modify the transformation and account for intensity scaling by redefining

$$s = T(r) = (L-1)\int_0^r p_r(w)dw \tag{6}$$

With this change, we get $s_0 = T(r_0) = 7p_r(r_0) = 1.33$.

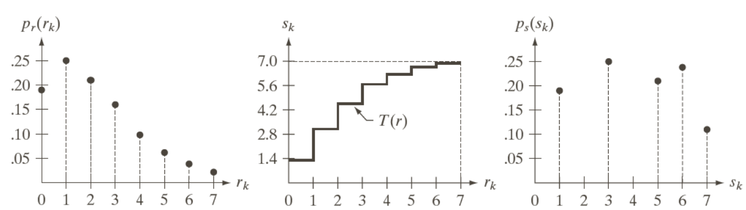
Example: contd.. map to closest integer values

similarly, $s_1 = 3.08, s_2 = 4.55, s_3 = 5.67, s_4 = 6.23, s_5 = 6.65, s_6 = 6.86, s_7 = 7$

Now map these values to the closest integer values:

$s_0 = 1.33 \to 1,\ s_1 = 3.08 \to 3,\ s_2 = 4.55 \to 5,\ s_3 = 5.67 \to 6,\ s_4 = 6.23 \to 6,\ s_5 = 6.65 \to 6,\ s_6 = 6.86 \to 6,\ s_7 = 7.00 \to 7$

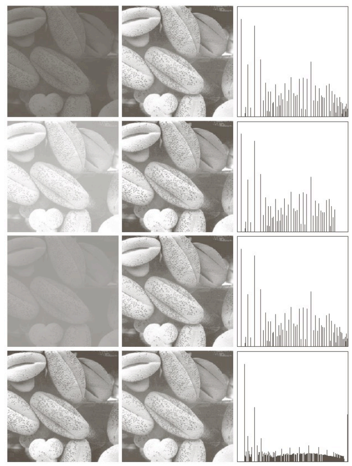
histograms



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Equalization: pictorial example



Histogram Specification

- Unlike equalization, here we specify the desired histogram. If the desired histogram is uniform, then it is the same as equalization. How do we do the mapping from a given image to an image with a specified distribution?
- Basically, map the given image histogram and the specified histopgram to the common space of equalized histograms. Then establish the correspondence between the intensities in this common space, and then invert it to get the desired image/histogram.

Things you should know

- Histogram Equalization. This is a pixel-level operation that maps a given intensity value to a new intensity level. The goal is to perform contrast stretching. The problem is formulated as one of taking the histogram of a given image and making that histogram uniform.
- Mean filtering or Average filtering. Here you smooth the image, usually go reduce the *noisyness* in the image. As the name indicates, you average the pixel values in a local neighborhood. An example filter is the one you already are very familiar with: Gaussian filtering, which does a weighted averaging over a given neighborhood. The center pixel value is replaced with the average value computed within the neighboorhood.
- Median filtering. Similar to mean filtering, except that you replace the pixel value with the median of the values in a given neighborhood. Note that median filtering is a non-linear operation. **You can not implement this in the Fourier domain, unlike mean filtering operations.**

15. Restoration and Wiener Filtering

3 Notes on Image Restoration and Wiener Filtering

A typical image observation model can be written as

$$g(x,y) = T[f(x,y)] + n(x,y) \tag{7}$$

where $f(x,y)$ is the image that is observed, $g(x,y)$ is the actual observation that is corrupted by the process $T[.]$ and additive noise $n(x,y)$.

Examples include motion blur (we discussed this in class), atmospheric turbulence (again, discussed in class), the imaging setup itself (optics related), CCD interactions, etc. Let us consider the motion blur again: The object $f(x,y)$ being imaged movesuniformly in the $x-$ direction at a velocity v. If the time exposure is T, the observed image can be written as

$$g(x,y) = \frac{1}{T}\int_0^T f(x - vt, y)\ dt = \frac{1}{\alpha_0}\int_0^{\alpha_0} f(x-\alpha, y)\ d\alpha \tag{8}$$

where $\alpha = vt$, $\alpha_0 = vT$. The above expression can be simplified to

$$g(x,y) = \frac{1}{\alpha_0}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} rect\left(\frac{\alpha}{\alpha_0} - \frac{1}{2}\right)\delta(\beta) f(x-\alpha, y-\beta)\ d\alpha\ d\beta \tag{9}$$

From the above equation, it is easy to conclude that the motion-blurred imaging process is shift invariant, with the Point Spread Function given by

$$h(x,y) = \frac{1}{\alpha_0} rect\left(\frac{x}{\alpha_0} - \frac{1}{2}\right)\delta(y) \tag{10}$$

Inverse Filtering

As the name indicates, inverse filtering restores the degradation due to "filtering". If you consider a linear model (convolution), then $g(x,y) = h(x,y) * f(x,y)$ and $G(u,v) = H(u,v)\ F(u,v)$. To retore the original $f(x,y)$, then, you filter the observation $g(x,y)$ with the inverse filter $H^I(u,v) = \frac{1}{H(u,v)}$. Note that this inverse filter does not exist if $H(u,v)$ has any zeros.

Pseudoinverse filter

This addresses the issues with possible zeros in $H(u,v)$ in constructing the inverse filter. A Pseudoinverse filter is defined as

$$H^-(u,v) = \begin{cases} \frac{1}{H(u,v)}, & H \neq 0 \\ 0, & H = 0 \end{cases} \tag{11}$$

$H^-(u,v)$ is also called the *generalized inverse* of $H(u,v)$. In practice, we set $H^-(u,v)$ to zero if $|H|$ is less than some small positive number ϵ.

Wiener Filtering

Inverse and Pseudoinverse filtering are sensitive to noise... they amplify the noise when for low values of $H(u,v)$. Wiener filtering addresses this issue, by accounting for noise effectly explicitly.

Assuming (without loss of generality) a zero-mean process for both the original signal and observed signal (and hence the underlying noise process), the mean square error between the original and the reconstructed signal is minimized by the conditional mean, i.e., given the observation from eq. 7, the best estimate $\hat{f}(x,y)$ that minimizes the error is given by

$$\hat{f}(x,y) = E\left\{f(x,y)|g(.)\right\} \tag{12}$$

This optimal estimate is, in general, non-linear and difficult to estimate as it requires the knowledge of joint distributions of the signal and observation model. Instead, we generally seek the *best linear estimate*, where the estimate is obtained as a linear combination of the observed data. Referring to your class slides, this estimate can be obtained using the orthogonality principle.

At this point, the best linear estimate may require that the "weighting kernel" $m(r,r')$ is spatially varying, refer to page 12.13 in your restoration slides lecture. However, if you further assume that $f(.)$ and $g(.)$ are jointly stationary random fields, then the corresponding filters simplify to spatially invariant filters of the form $m(r-r')$. This is because, for stationary fields, the autocorrelation function is a function of only the difference vectors $(r-r')$ and not absolute spatial locations (r,r'). This leads us to the final derivation for the wiener filter in terms of the spectral density functions

$$M(u,v) = \frac{S_{fg}(u,v)}{S_{gg}(uv)} \tag{13}$$

or

$$M(u,v) = \frac{H^*(u,v)S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{nn}(u,v)} \tag{14}$$

Note that this Wiener filter reduces to the inverse filter when the noise level is insignificant, i.e., $S_{nn} \approx 0$.

Question: What changes you need to make to the Wiener filter if the various quantities are not zero-mean? Say the noise $n(x,y)$ has a mean $\mu_n(x,y)$ and the true signal $f(x,y)$ has a mean $\mu_f(x,y)$.

To answer this, note that our observation model is $g(x,y) = (h*f)(x,y) + n(x,y)$. Subtracting the known mean out, define $f'(x,y) = f(x,y) - \mu_f(x,y)$ and $n'(x,y) = n(x,y) - \mu_n(x,y)$. This results in the zero mean model

$$g'(x,y) = h(x,y) * f'(x,y) + n'(x,y) \tag{15}$$

To get to g', assuming the mean of the original image and the noise process are known, we first subtract the mean value of g from the observed signal g. Note that this mean value, $\mu_g(x,y) = h(x,y) * \mu_f(x,y) + \mu_n(x,y)$. Then the Wiener filter is designed to give the zero mean estimate of f, so after you filter with the

Wiener filter on g', you add the signal mean back. In the frequency domain, these can be written as (and dropping the frequency index (u, v), for notational convenience)

$$\hat{F} = M(G - M_g) + M_f \tag{16}$$

where $M_g(u,v) = \mathcal{F}\{\mu_g(x,y)\}$. Substituting for M_g in terms of the known quantities, we get the final expression for the restored non-zero mean signal as

$$\hat{F} = MG + \frac{S_{nn}}{|H|^2 S_{ff} + S_{nn}} M_f - M M_n \tag{17}$$