Chapter 9

Image Restoration: Basic

Contents

Introduction **9.2**
Problem formulation 9.3
Degradation Estimation 9.4
Performance measures 9.6
Reduction of additive random noise **9.8**
Wiener filter 9.8
Adaptive image processing 9.8
Adaptive Wiener filter 9.9
Wavelet-based image denoising 9.12
Reduction of image blurring **9.12**
Inverse filtering 9.12
Blind deconvolution 9.12
Reducing blur and noise **9.13**
Wiener filter 9.13
Reduction of signal-dependent noise 9.14
Temporal filtering for image restoration **9.15**
Frame averaging 9.15
Motion-compensated image restoration 9.15
Color images 9.15
Summary **9.16**

(Related to Ch. 9 of Lim.)

Introduction

Entire books are devoted to this problem [1–5].

Example. Hubble space telescope. (Spherical lens aberration causes space-variant PSF, requiring iterative restoration methods.) Confocal microscopy. Photograph of moving object with slow shutter.

Overview

Organized by types of degradation.
- *9.2* additive random noise
- *9.3* blurring
- *9.4* blurring and noise
- *9.5* signal-dependent noise
- *9.6* multiple frames

MATLAB commands related to image restoration include the following. Of these, we will focus in particular on the adaptive Wiener filter for noise reduction `wiener2`.

```
deconvblind    - Deblur image using blind deconvolution.
deconvlucy     - Deblur image using Lucy-Richardson method.
deconvreg      - Deblur image using regularized filter.
deconvwnr      - Deblur image using Wiener filter.
wiener2        - Perform 2-D adaptive noise-removal filtering.
```

Of course LSI restoration methods are enabled by MATLAB's filtering commands, and iterative restoration methods are easily implemented using all of MATLAB's matrix computation routines.

Problem formulation

- Previously we considered **image enhancement**, where processing was performed on an image to highlight certain features of the image. All this processing was done assuming that the initial image was a fairly accurate representation of the objects of interest. Enhancement methods are *ad hoc* usually assuming no model for the degradation or object. Evaluation is typically qualitative.
- In this chapter we consider the problem of **image restoration**, where the image has been degraded prior to entering our image processing system. The goal of an image restoration algorithm is to generate an estimate of the original picture prior to the degradation.
- There is some overlap between image restoration and image enhancement. Often image enhancement algorithms, such as median filtering, can be used to restore degraded images. However, the term image restoration is usually associated with minimizing, or even removing, image artifacts due to *blurring* and noise. Restoration methods are usually based on explicit models and evaluated quantitatively (*e.g.*, MSE).
- There are multiple sources of blurring: camera motion, temporal aliasing, out of focus lens, etc. In general, the algorithms used to reduce the effects of blurring are specific to the source of blurring.
- There are multiple sources of noise: additive Johnson noise, bit loss in communication, multiplicative noise (*e.g.*, speckle), etc. Again, the sources of noise usually dictate the type of processing performed.
- In actual systems the degradation often occurs *before* the digitizing system, *i.e.*, in the following model.

$$\underbrace{f(x,y) \to \boxed{\text{Degradation}} \to g(x,y) \to \boxed{\text{Digitizer}} \to}_{\text{model}} g[n,m] \to \boxed{\text{Restoration}} \to \hat{f}[n,m] \to \boxed{\text{Display}} \to \hat{f}(x,y)\,.$$

Thus the ultimate goal could be to recover $f(x,y)$ from $g[n,m]$. The above model is sometimes called a **continuous-to-discrete** degradation. Although the model shown above is often theoretically appropriate, for simplicity, many image restoration methods are developed using a simpler *discrete-to-discrete* degradation model in which the degradation occurs after the digitization process, as shown below:

$$\underbrace{f(x,y) \to \boxed{\text{Digitizer}} \to f[n,m] \to \boxed{\text{Degradation}} \to}_{\text{model}} g[n,m] \to \boxed{\text{Restoration}} \to \hat{f}[n,m] \to \boxed{\text{Display}} \to \hat{f}(x,y)\,.$$

Everything to the left of $g[n,m]$ *in the above diagrams is a* model *only.* It has been said that "all models are wrong, but some models are useful," and this remark can apply to either of the above models.

If an image restoration procedure is accurate, then the processed image

$$\hat{f}[n,m] \approx f[n,m]\,.$$

Assuming a discrete-space degradation model facilitates discrete-space algorithms for image restoration. For many forms of degradation, this assumption is sufficiently accurate.

One can also use a discrete-to-discrete model but include a down-sampler in the model for a closer approximation to the continuous-to-discrete model:

$$\underbrace{f(x,y) \to \boxed{\text{Digitizer}} \to f[n,m] \to \boxed{\text{Degradation}} \to \boxed{\text{Down sampling}} \to}_{\text{model}} g[n,m] \to \boxed{\text{Restoration}} \to \hat{f}[n,m]\,.$$

Of course $f[n,m]$ is never known in practice, but we often know something about the characteristics of $f[n,m]$ that can help design the restoration method. For example, we might assume $f[n,m]$ is the realization of a random process with certain statistical properties. Under such models we have the possibility of designing a restoration system that is optimal in some sense, such as minimizing the MSE $\mathsf{E}\left[\left|\hat{f}[n,m] - f[n,m]\right|^2\right]$.

9.1

Degradation Estimation

Before some degradation can be removed, we must know something about its properties. Sometimes we know its properties *a priori*. In most circumstances, however, we do not know the properties fully. Consequently, a common first step in any image restoration procedure is to estimate the degradation itself so that procedures for "optimal" removal can then be designed.

Two main approaches to characterizing degradations.

- Extract information about the degradation processes from the degraded image itself.
  If an image is degraded by additive noise, then we can estimate the power spectrum or the probability density function of the random noise process from intensity fluctuations within a uniform brightness region, such as the sky or the background surrounding the brain in a brain image.
  Estimating a noise power spectrum makes an additional implicit assumption (beyond additivity) of the noise.
  What is it? **??**
  If an image is blurred by an out of focus lens such that the degraded image is approximately related to $f[n,m]$ by

$$g[n,m] \approx f[n,m] ** b[n,m],$$

  then we can estimate the blurring function $b[n,m]$ if the object contains an impulse (or something close to an impulse), that is $f[n,m] \approx \delta_2[n-n_0, m-m_0]$ somewhere in the image.
  Also, if there is a uniform speckle field (only true for coherent imaging systems), then the blurring function can be estimated by the 2D correlation function in the neighborhood of the speckle field.
- Another approach to degradation characterization is to directly model the degradation (based on studying the degradation mechanisms) and then to estimate parameters of the model from the image or from separate "calibration" measurements.

Example. Consider an analog image $f_a(x,y)$ that is blurred by planar motion during photographic exposure. A simple model for a photograph of a flat **stationary scene** taken by a moving camera is:

$$g_a(x,y) = \frac{1}{T}\int_{-T/2}^{T/2} f_a(x-x_0(t), y-y_0(t))\,dt = f_a(x,y) ** \frac{1}{T}\int_{-T/2}^{T/2} \delta_2(x-x_0(t), y-y_0(t))\,dt, \qquad (9\text{-}1)$$

where $x_0(t)$ and $y_0(t)$ represent the horizontal and vertical displacements of the camera relative to its position at time $t=0$, assuming $x_0(0) = y_0(0) = 0$, and T is the total exposure duration.

Does the degradation in (9-1) correspond to an LSI system? **??**

The PSF of this motion blur and its corresponding frequency response are given by:

$$b_a(x,y) = \frac{1}{T}\int_{-T/2}^{T/2} \delta_2(x-x_0(t), y-y_0(t))\,dt \overset{\mathcal{F}}{\longleftrightarrow} B_a(\nu_X,\nu_Y) = \frac{1}{T}\int_{-T/2}^{T/2} e^{-\imath 2\pi(\nu_X x_0(t) + \nu_Y y_0(t))}\,dt\,.$$

Example. If there is linear motion in the x direction such that $x_0(t) = kt$ and $y_0(t) = 0$, then $B_a(\nu_X,\nu_Y)$ reduces to

$$B_a(\nu_X,\nu_Y) = \operatorname{sinc}(k\nu_X T)\,.$$

(Note that k must have units distance/time, which balances above.) In the space domain,

$$b_a(x,y) = \frac{1}{kT}\operatorname{rect}\left(\frac{x}{kT}\right) \boxed{??}.$$

What is missing from the above? **??**

Suppose we are given a digital photograph corrupted by (assumed known) motion blur and we wish to perform discrete-space restoration.

What should the corresponding discrete-to-discrete blur function $b[n,m]$ for this model?

If $f[n,m] = f_a(n\Delta_X, m\Delta_Y)$, then does there exist a discrete impulse response $b[n,m]$ such that

$$g[n,m] = g_a(n\Delta_X, m\Delta_Y) = f[n,m] ** b[n,m]?$$

Digital restoration algorithms require such a $b[n,m]$, so this is an important practical point to resolve!

In particular, it may be reasonable to assume that $f_a(x,y)$ is band-limited, but whether $b(x,y)$ is band-limited may be less clear.

Since the system is LSI, if $f_a(x,y)$ is bandlimited, then so is $g_a(x,y)$. Thus, working in the frequency domain we have

$$\begin{aligned} g[n,m] \overset{\text{DSFT}}{\longleftrightarrow} G(\omega_X,\omega_Y) &= \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \frac{1}{\Delta_X\Delta_Y} G_a\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right) \\ &= \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \frac{1}{\Delta_X\Delta_Y} F_a\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right) B_a\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right) \\ &= F(\omega_X,\omega_Y)\, B(\omega_X,\omega_Y) \end{aligned}$$

where (assuming $f_a(x,y)$ is bandlimited and adequately sampled):

$$\begin{aligned} F(\omega_X,\omega_Y) &= \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \frac{1}{\Delta_X\Delta_Y} F_a\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right) \\ B(\omega_X,\omega_Y) &= \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} B_a\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right) \operatorname{rect}_2(\omega_X/2\pi - k, \omega_Y/2\pi - l)\,. \end{aligned}$$

Normally a sum of products does not equal the products of sums. Why does it work here? **??**

Thus, in the bandlimited case, we have the following *exact* discrete-to-discrete model:

$$g[n,m] = f[n,m] ** b[n,m]$$

where the required blur $b[n,m]$ has the following frequency response

$$b[n,m] \overset{\text{DSFT}}{\longleftrightarrow} B(\omega_X,\omega_Y) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \frac{1}{\Delta_X\Delta_Y} H\left(\frac{\omega_X/2\pi - k}{\Delta_X}, \frac{\omega_Y/2\pi - l}{\Delta_Y}\right)$$

where

$$H(\nu_X,\nu_Y) = B_a(\nu_X,\nu_Y)\,\Delta_X\Delta_Y \operatorname{rect}_2(\Delta_X\nu_X, \Delta_Y\nu_Y)\,.$$

So $b[n,m]$ corresponds to samples of $h(x,y)$, where

$$h(x,y) = b_a(x,y) ** \operatorname{sinc}_2\left(\frac{x}{\Delta_X}, \frac{y}{\Delta_Y}\right).$$

Of course if $b_a(x,y)$ is already appropriately bandlimited, then the convolution with the sinc has no effect.

The book describes $B(\omega_X,\omega_Y)$ as "an aliased version of $B_a(\nu_X,\nu_Y)$." Do you agree? **??**

Given this discrete approximation, coefficients describing the degradation model must be obtained. For the example stated above of linear motion only along the x direction, we need "only" to estimate k (relative camera velocity) from the original image. That is, the total blurring function can be accurately modeled *if* the parameter k can be reasonably estimated from the original image. The motion parameter can be obtained from alternate information or by estimating the blur of a known object in the photograph (*e.g.*, from an appropriately oriented object boundary that is known to be sharp but appears blurred).

Performance measures

To quantify both the degree of degradation as well as the improvements due to restoration algorithms, various performance measures are used in the literature. Here are some examples.

- The **mean square error** (**MSE**) between a restoration $\hat{f}[n, m]$ and an original $N \times M$ image $f[n, m]$ is the per-pixel average error energy:

$$\mathsf{MSE} = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left| \hat{f}[n, m] - f[n, m] \right|^2 .$$

  Note that this is a *spatial* average, not an ensemble average.
- If we have a statistical model for the images, *i.e.*, we treat $f[n, m]$ as random process, then often we define **MSE** in terms ensemble averages:

$$\mathsf{MSE} = \mathsf{E}\left[\left| \hat{f}[n, m] - f[n, m] \right|^2 \right] .$$

  Note that this MSE could be a function of pixel location n,m, but typically we use this MSE in the context of WSS random process models, in which case the MSE is not a function of n,m.
- A slight inconvenience of MSE is that its units are the square of the original image units. In photographic imaging, where the intensities typically are arbitrary units, this is a minor issue. But in applications like X-ray CT, where the pixel values have physically meaningful units, it can be preferable to use error measures that are expressed in those very units. One simple solution is to take the square root of MSE, known as the **root mean-squared error** (**RMSE**):

$$\mathsf{RMSE} = \sqrt{\mathsf{MSE}}.$$

- In cases where the image units are arbitrary, it is preferable to report error measures that are invariant to the choice of units by normalizing. There are a multiple definitions of **normalized mean square error** (**NMSE**) in the literature, so when reporting results it is wise to specify which version was used. Examples for single realizations are:

$$\mathsf{NMSE} = \frac{\mathsf{MSE}}{\frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} |f[n, m]|^2}, \ \mathsf{NMSE} = \frac{\mathsf{MSE}}{\frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left|f[n, m] - \bar{f}\right|^2}, \text{ where } \bar{f} \triangleq \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n, m],$$

  and for cases with random process models:

$$\mathsf{NMSE} = \frac{\mathsf{MSE}}{\mathsf{E}\left[|f[n, m]|^2 \right]}, \qquad \mathsf{NMSE} = \frac{\mathsf{MSE}}{\mathsf{Var}\{f[n, m]\}}.$$

  Lim defines the **normalized mean square error** (**NMSE**) between the original image $f[n, m]$ and the restored image $\hat{f}[n, m]$ as follows:

$$\mathsf{NMSE}\left\{ f[n, m], \hat{f}[n, m] \right\} \triangleq \frac{\mathsf{Var}\left\{ f[n, m] - \hat{f}[n, m] \right\}}{\mathsf{Var}\{f[n, m]\}} \cdot 100\%.$$

  Lim notes that using the variance yields invariance to DC shifts (bias). But in quantitative applications, such biases are important.
- No matter which NMSE definition is used, usually we define the **SNR** as its reciprocal, possibly in dB:

$$\mathsf{SNR} = \frac{1}{\mathsf{NMSE}}, \qquad \mathsf{SNR} = 10 \log_{10} \frac{1}{\mathsf{NMSE}}.$$

- The **peak SNR** or **PSNR** is also frequently reported, often in dB:

$$\text{PSNR} \triangleq 10 \log_{10} \frac{\max_{(n,m)} |f[n, m]|^2}{\mathsf{MSE}}.$$

- Another (less common) option is to modify the MSE definition to make it invariant to amplitude scale factors:

$$\mathsf{MSE} = \min_{\alpha} \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left| \alpha \, \hat{f}[n, m] - f[n, m] \right|^2 ,$$

or to make it invariant to spatial shifts:

$$\mathsf{MSE} = \min_{n',m'} \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left| \hat{f}[n-n', m-m'] - f[n,m] \right|^2 .$$

- All of the above are "squared error" performance measures, which are convenient for analysis. But worst-case errors and ℓ_1 errors can also be of interest:

$$\max_{n,m} \left| \hat{f}[n,m] - f[n,m] \right|, \qquad \frac{1}{NM} \left\| \hat{f} - f \right\|_1 = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left| \hat{f}[n,m] - f[n,m] \right| .$$

Clearly there are a variety of possible performance measures.

9.2

Reduction of additive random noise

I consider this topic to be **denoising** not **image restoration**...

Additive random noise model:

$$g[n,m] = f[n,m] + v[n,m],$$

where $v[n,m]$ is additive noise, usually assumed to be **independent** of the signal $f[n,m]$, unless otherwise specified.

Example. Thermal noise in CCD detector electronics, or (in some cases) amplitude quantization noise.

(Caution: quantization noise can be signal dependent.)

Note that there is no blurring considered in this model.

9.2.1

Wiener filter: effect of non-zero mean

As derived previously, if the signal and noise are zero mean and WSS, the Wiener filter gives minimum MSE over all LSI methods, where

$$\hat{f}[n,m] = h[n,m] ** g[n,m], \text{ where } h[n,m] \overset{\text{DSFT}}{\longleftrightarrow} H(\omega_X, \omega_Y) = \frac{P_f(\omega_X, \omega_Y)}{P_f(\omega_X, \omega_Y) + P_v(\omega_X, \omega_Y)},$$

for uncorrelated signal and noise.

Because $H(0,0) = \frac{P_f(0,0)}{P_f(0,0)+P_v(0,0)} < 1$, if the signal mean μ_f is nonzero, then the usual (**linear**) Wiener filter would be biased:

$$\mathsf{E}\left[\hat{f}[n,m]\right] = \mathsf{E}[h[n,m] ** g[n,m]] = h[n,m] ** \mathsf{E}[g[n,m]] = h[n,m] ** \mu_f = \mu_f \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h[n,m] = \mu_f \, H(0,0) < \mu_f.$$

Thus, for signals with (known!) nonzero mean μ_f, it is natural to use the following **affine** estimator:

$$\hat{f}[n,m] = \mu_f + h[n,m] ** (g[n,m] - \mu_f).$$

A global known mean may be unrealistic. A more realistic model might be $f[n,m] = \mu_f[n,m] + f_0[n,m]$, where $f_0[n,m]$ is zero-mean and WSS, and $\mu_f[n,m]$ is "known" or obeys a model with some unknown parameters. This is sometimes called a **Wold decomposition**. Now the "affine Wiener filter" problem is to find $\mu_f[n,m]$ and $h[n,m]$ that minimize MSE.

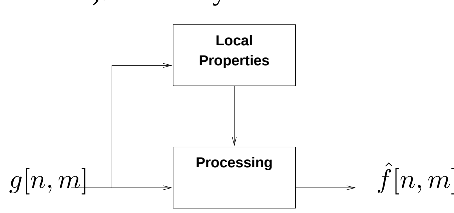
9.2.3

Adaptive image processing

The Wiener derivation assumes WSS signal and noise. For Gaussian noise, the resulting MMSE estimation method is LSI. The WSS assumption may be reasonable for noise in many circumstances, but is unrealistic for images (signals), since often the local signal spectrum changes drastically between different image regions.

In **adaptive methods**, one deliberately changes the behavior of the "filter" in different image regions according to some criteria.

Basically, to reduce noise we would like to smooth a lot where the true signal is uniform (low frequency), and smooth less where the true signal varies rapidly (near edges in particular). Obviously such considerations are application dependent.



One can apply adaptive methods either **pixel by pixel** or **block by block**. The pixel by pixel methods generally require more computation. However, the block by block methods can cause abrupt changes in image intensity between blocks, called the **blocking effect**. A common and effective method for reducing the blocking effect is to use **overlapping blocks**. This increases computation somewhat, but not as much as full pixel by pixel methods.

9.2.4

Adaptive Wiener filter for the additive noise model: $g[n,m] = f[n,m] + v[n,m]$.

The idea: use a *shift variant* filter whose width depends on the local image features. In mathematical form:

$$\hat{f}[n,m] = \sum_{n'} \sum_{m'} h[n,m;n',m';g] \, g[n',m'] \, .$$

There are many ways to choose such adaptive filter. For the (now classic) adaptive Wiener filter, one assumes that *locally*

$$f[n,m] = m_f + \sigma_f \, u[n,m]$$

where m_f is the local mean (average of neighboring pixels), σ_f is the local standard deviation of $f[n,m]$, and $u[n,m]$ represents local uncorrelated signal variations having **unit variance**, *i.e.*, $\mathsf{R}_u[n,m] = \delta_2[n,m]$. Then defining the zero-mean signal $f' = f - m_f$, we have $P_{f'}(\omega_X, \omega_Y) = \sigma_f^2$ so the (linear) Wiener filter is

$$H(\omega_X, \omega_Y) = \frac{P_{f'}(\omega_X, \omega_Y)}{P_{f'}(\omega_X, \omega_Y) + P_v(\omega_X, \omega_Y)} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2} = \frac{1}{1 + \sigma_u^2/\sigma_f^2} \text{ and } h[n,m] = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_u^2} \, \delta_2[n,m] \, .$$

Thus (using the affine form of the Wiener filter that accounts for nonzero mean):

$$\hat{f}[n,m] = m_f + h[n,m] ** (g[n,m] - m_f) = m_f + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_u^2}(g[n,m] - m_f) = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_u^2} \, g[n,m] + \frac{\sigma_u^2}{\sigma_f^2 + \sigma_u^2} m_f .$$

In practice, we apply the above principle *locally* using estimates of the local mean and variance:

$$\boxed{\hat{f}[n,m] = \frac{\hat{\sigma}_f^2[n,m]}{\hat{\sigma}_f^2[n,m] + \sigma_u^2} \, g[n,m] + \frac{\sigma_u^2}{\hat{\sigma}_f^2[n,m] + \sigma_u^2} \, \hat{m}_f[n,m],}$$

which is a weighted combination of the (estimated) **local mean** $\hat{m}_f$ and **local contrast** $g[n,m] - m_f$, where

$$\hat{m}_f[n,m] = \hat{m}_g = \frac{1}{(2M+1)^2} \sum_{k=-M}^{M} \sum_{l=-M}^{M} g[n-k, m-l] \quad \text{(a \textbf{moving average}!)}$$

$$\hat{\sigma}_g^2[n,m] = \frac{1}{(2M+1)^2 - 1} \sum_{k=-M}^{M} \sum_{l=-M}^{M} \left(g[n-k, m-l] - \hat{m}_f[n,m]\right)^2$$

$$\hat{\sigma}_f^2[n,m] = \max\{\hat{\sigma}_g^2[n,m] - \sigma_u^2, 0\}.$$

(The "-1" above is perhaps unessential, but is motivated by the unbiasedness of the usual sample standard deviation.)

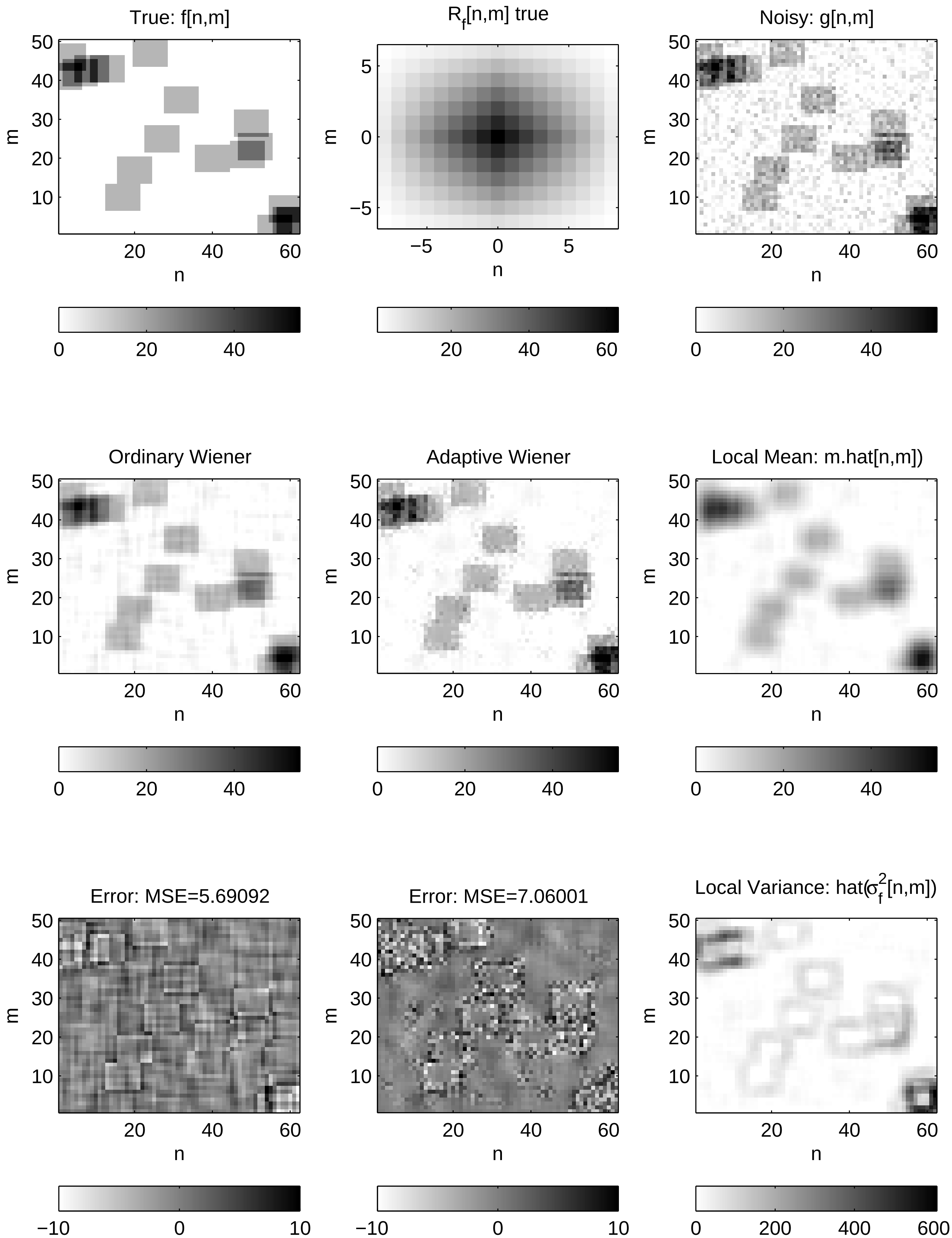
Note that with the **rectangular window** $w[n,m] = \begin{cases} \frac{1}{(2M+1)^2-1}, & |k|,|l| \le M \\ 0, & \text{otherwise,} \end{cases}$ we have

$$\begin{aligned} \hat{\sigma}_g^2[n,m] &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} w[k,l] \left(g[n-k, m-l] - \hat{m}_f[n,m]\right)^2 \\ &= w[n,m] ** g^2[n,m] - 2 \, \hat{m}_f[n,m] \cdot (w[n,m] ** g[n,m]) + \hat{m}_f^2[n,m] \, \frac{(2M+1)^2}{(2M+1)^2 - 1} . \end{aligned}$$

- This is another **two-channel** processing method.
- Reduces blurring relative to the LSI Wiener method.
- There is still the issue of choosing neighborhood size.

Unfortunately, the adaptive Wiener filter still produces noisy edges, as illustrated in the following example.

Why does the adaptive Wiener filter perform poorly in this case? **??**


True: f[n,m]
m
n
Rf[n,m] true
Noisy: g[n,m]
Ordinary Wiener
Adaptive Wiener
Local Mean: m.hat[n,m])
Error: MSE=5.69092
Error: MSE=7.06001
Local Variance: hat(σf2[n,m])

9.2.7 Edge-sensitive adaptive image restoration

The preceding example showed that the isotropic characteristic of the classical adaptive Wiener filter yields suboptimal results near image edges. An alternative is to apply a 1D adaptive Wiener filter in x, then in y, then perhaps at $\pm 45°$. This "separable" approach saves computation, and often works better than the 2D approach!

Example. For horizontal filter, we simply use a horizontal row of pixels to estimate the local mean and variance, as follows:

$$\hat{m}_f[n,m] = = \frac{1}{2M+1} \sum_{k=-M}^{M} g[n-k,m] \text{ (a 1D \textbf{moving average}!)}$$

$$\hat{\sigma}_g^2[n,m] = \frac{1}{(2M+1)-1} \sum_{k=-M}^{M} \left(g[n-k,m] - \hat{m}_f[n,m]\right)^2.$$

9.2.5 Adaptive restoration based on the noise visibility function

This is yet another variation on the Wiener filter that adapts based on a quantitative measure of how visible a given type of noise would be to a human viewer.

Tuning parameters:
- Width of the local impulse response
- Weighting between raw image $g[n,m]$ and smoothed image

9.2.6 Short-space spectral subtraction

Consider a small, say $M \times M$, spatial window. Within that $M \times M$ window:

$$g[n,m] = f[n,m] + v[n,m] \overset{\text{DSFT}}{\longleftrightarrow} G(\omega_X,\omega_Y) = F(\omega_X,\omega_Y) + V(\omega_X,\omega_Y),$$

and assuming zero-mean uncorrelated noise that is uncorrelated with the true image:

$$\mathsf{E}\left[|G(\omega_X,\omega_Y)|^2\right] = \mathsf{E}\left[|F(\omega_X,\omega_Y)|^2\right] + \mathsf{E}[F(\omega_X,\omega_Y)\,V^*(\omega_X,\omega_Y)] + \mathsf{E}[F^*(\omega_X,\omega_Y)\,V(\omega_X,\omega_Y)] + \mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right]$$
$$= \mathsf{E}\left[|F(\omega_X,\omega_Y)|^2\right] + \mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right].$$

Since $\mathsf{E}\left[|F(\omega_X,\omega_Y)|^2\right] = \mathsf{E}\left[|G(\omega_X,\omega_Y)|^2\right] - \mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right]$, one way to estimate $|F(\omega_X,\omega_Y)|$ is

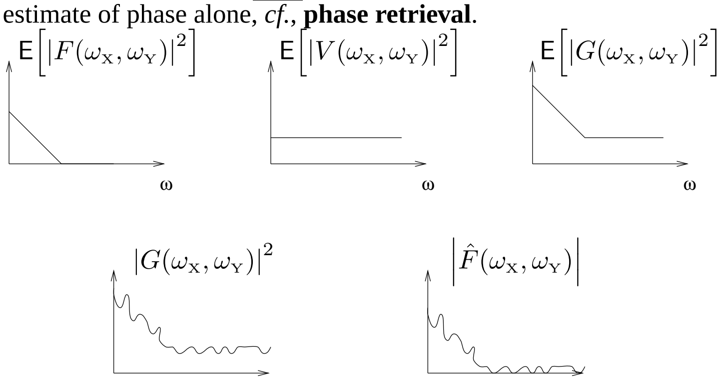
$$|\widehat{F(\omega_X,\omega_Y)}| = \sqrt{\max\left\{|G(\omega_X,\omega_Y)|^2 - \mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right], 0\right\}},$$

where we assume $\mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right]$ is known (usually a constant). An estimate for $F(\omega_X,\omega_Y)$ is thus:

$$\hat{F}(\omega_X,\omega_Y) = \sqrt{\max\left\{|G(\omega_X,\omega_Y)|^2 - \mathsf{E}\left[|V(\omega_X,\omega_Y)|^2\right], 0\right\}}\, e^{\imath \angle G(\omega_X,\omega_Y)}.$$

Why is using the phase of $G(\omega_X,\omega_Y)$ reasonable? **??**

However, this is not an optimal estimate of phase alone, *cf.*, **phase retrieval**.



Wavelet-based image denoising

Take wavelet transform, shrink small coefficients towards zero. Leave larger coefficients unaltered. Hard vs soft thresholding [6]. [7].

9.3

Reduction of image blurring

Next we consider the following dubious model:

$$g[n,m] = b[n,m] ** f[n,m] \,.$$

Notice the lack of a noise term in this model, yet within 3 paragraphs of this section in Lim the word "noise" appears!

There are two categories of deblurring problems: when $b[n,m]$ is known, and when it is unknown (or partially known).

9.3.1

Inverse filtering

$$G(\omega_X,\omega_Y) = B(\omega_X,\omega_Y)\,F(\omega_X,\omega_Y), \text{ so } \hat{F}(\omega_X,\omega_Y) = \begin{cases} \dfrac{G(\omega_X,\omega_Y)}{B(\omega_X,\omega_Y)}, & B(\omega_X,\omega_Y) \neq 0, \\ ?, & \text{otherwise.} \end{cases}$$

This method is *very* sensitive to noise, since $B(\omega_X,\omega_Y)$ usually approaches zero for high spatial frequencies ***(Picture)*** .

There are *ad hoc* fixes to this solution, but the problem is the *model*, not the *solution*. As a general rule, it is preferable to replace an artificial model with a more complete model (*e.g.*, by including a noise term) and rederive a better solution than to try to "fix up" a correct solution to an incorrect model.

A trivial iteration

The book also describes an **iterative algorithm** for image deblurring. The particular algorithm described ends up being exactly equivalent to filtering the image using a filter with frequency response:

$$H(\omega_X,\omega_Y) = \frac{1-(1-\lambda\,B(\omega_X,\omega_Y))^{k+1}}{B(\omega_X,\omega_Y)} = \frac{1}{B(\omega_X,\omega_Y)}\left[1 - \sum_{j=0}^{k+1}\binom{k+1}{j}(-\lambda\,B(\omega_X,\omega_Y))^j\right]$$

$$= \frac{1}{B(\omega_X,\omega_Y)}\sum_{j=1}^{k+1}\binom{k+1}{j}(-\lambda\,B(\omega_X,\omega_Y))^j = \sum_{j=1}^{k+1}\binom{k+1}{j}(-\lambda\,B(\omega_X,\omega_Y))^{j-1}.$$

Thus for this particular example, there is absolutely no point to using an iterative algorithm: just apply the corresponding filter and you get the same results with much less computation.

If however, the blur is shift-variant, or if other constraints (such as $f[n,m] \geq 0$) are desired, *then* iterative algorithms are useful.

9.3.2

Blind deconvolution

All of the methods described herein assume that $b[n,m]$ is known. In some applications $b[n,m]$ may be unavailable or difficult to determine.

Trying to determine both $b[n,m]$ and $f[n,m]$ from $g[n,m] = b[n,m] ** f[n,m]$ is a severely under-determined problem. To illustrate, if $(f[n,m], b[n,m])$ is one solution, then $(b[n,m], f[n,m])$ is an equally good solution as far as the data is concerned. So **blind deconvolution** methods work best if considerable prior information about $b[n,m]$ is available.

- Support constraints
- Nonnegativity
- Parametric forms for $b[n,m]$

9.4

Reducing blur and noise

Now consider the following (more realistic) measurement model:

$$g[n,m] = b[n,m] ** f[n,m] + v[n,m] = (b ** f)[n,m] + v[n,m]\,.$$

To recover $f[n,m]$ from $g[n,m]$ we must both reduce the effects of noise $v[n,m]$ and "deconvolve" the effects of the blur $b[n,m]$.

Two-step methods are a simple approach: first reduce noise, then try a blur removal method. This type of approach is generally not very good since the blur removal methods described in the previous section assume noiseless data, whereas noise reduction methods can at best *reduce* noise, not *eliminate* it.

Wiener filter

One can derive ***(Problem 9.15)*** the following Wiener filter for this problem if $f[n,m]$ and $v[n,m]$ are WSS random processes:

$$H(\nu_X,\nu_Y) = \frac{P_f(\omega_X,\omega_Y)\,B^*(\omega_X,\omega_Y)}{P_f(\omega_X,\omega_Y)\,|\,B(\omega_X,\omega_Y)\,|^2 + P_v(\omega_X,\omega_Y)} = \left(\frac{P_f(\omega_X,\omega_Y)\,|\,B(\omega_X,\omega_Y)\,|^2}{P_f(\omega_X,\omega_Y)\,|\,B(\omega_X,\omega_Y)\,|^2 + P_v(\omega_X,\omega_Y)}\right)\frac{1}{B(\omega_X,\omega_Y)}.$$

One can view this filter as a cascade of a noise reduction system (the Wiener filter derived earlier) and a deblurring system (the $1/B(\omega_X,\omega_Y)$ part). However, this formulation does *not* suffer from the instabilities of the naive $1/B(\omega_X,\omega_Y)$ inverse filter, since the $B(\omega_X,\omega_Y)$ in the numerator cancels out the $1/B(\omega_X,\omega_Y)$ term.

Matrix formulation

One can also derive a Wiener-like estimation method for the more general linear model:

$$\boldsymbol{g} = \boldsymbol{B}\boldsymbol{f} + \boldsymbol{v},$$

where $\boldsymbol{f}$ and $\boldsymbol{v}$ need not be WSS.

The MMSE affine estimator of $\boldsymbol{f}$ (see Section 6.7 of [8]) is:

$$\hat{\boldsymbol{f}} = \mathsf{E}[\boldsymbol{f}] + \mathsf{Cov}\{\boldsymbol{f},\boldsymbol{g}\}\,\mathsf{Cov}\{\boldsymbol{g}\}^{-1}\,(\boldsymbol{g} - \mathsf{E}[\boldsymbol{g}])\,.$$

In the usual case where the signal and noise are uncorrelated and the noise is zero mean, then

$$\hat{\boldsymbol{f}} = \mathsf{E}[\boldsymbol{f}] + \mathsf{Cov}\{\boldsymbol{f}\}\,\boldsymbol{B}'\,[\boldsymbol{B}\,\mathsf{Cov}\{\boldsymbol{f}\}\,\boldsymbol{B}' + \mathsf{Cov}\{\boldsymbol{v}\}]^{-1}\,(\boldsymbol{g} - \boldsymbol{B}\,\mathsf{E}[\boldsymbol{f}])\,.$$

We mentioned previously some of the challenges in finding the power spectrum of signals and noise even in the WSS. Finding the entire covariance matrix $\mathsf{Cov}\{\boldsymbol{f}\}$ is usually even more impractical in imaging problems!

This section of the text* barely *scratches the surface of the extensive literature on image restoration methods based on sound statistical principles. We will discuss better approaches later as time permits.

9.5 Reduction of signal-dependent noise

(Arises for nonlinear imaging systems, nonnegativity considerations, non-Gaussian noise, etc.)
A more general degradation "model" is:

$$g[n,m] = D[f[n,m]]. \tag{9.74a}$$

If we define

$$e[n,m] \triangleq g[n,m] - f[n,m],$$

then we can write (9.74a) in the following form:

$$g[n,m] = f[n,m] + e[n,m]\,. \tag{9.74a}$$

If the statistics of $e[n,m]$ are *not* a function of the signal $f[n,m]$, then we can describe $e[n,m]$ as **additive, signal-independent noise**. If the properties of $e[n,m]$ depend on the signal, then we say the noise is **signal dependent** and the additive expression (9.74a) is not very useful, and even misleading (and should be avoided in my opinion). Even the "general" form (9.74a) is not really general enough for some types of measurement noise.

Example. Binary imaging: $\mathsf{P}\{g[n,m]=1\} = 1 - \exp(-f[n,m])\,.$

Example. In problems involving photons, the measurement statistics are often Poisson. For such problems we often assume:

$$g[n,m] \sim \mathsf{Poisson}\{\alpha\, f[n,m] + \gamma\},$$

where α and γ are nonnegative constants. This notation is shorthand for saying that $g[n,m]$ is a Poisson random variable with mean $\alpha\, f[n,m] + \gamma$, *i.e.*,

$$\mathsf{P}\{g[n,m] = k\} = (\alpha\, f[n,m] + \gamma)^k\, \mathrm{e}^{-(\alpha\, f[n,m] + \gamma)} / k!, \;\; k = 0, 1, \ldots. \tag{9.74a}$$

The preceding formula is rigorous and unambiguous; expressions like (9.74a) and (9.74a) are often seen in (sloppy) papers on problems with Poisson "noise," but are imprecise and should be avoided.

Aside from the notation issues, the problem becomes: how do we recover $f[n,m]$ from $g[n,m]$ in the presence of signal-dependent noise? The answer of course depends greatly on the *type* of degradation.

The MAP or penalized-likelihood framework described elsewhere easily accommodates signal-dependent noise, provided one can specify a statistical model $p(g|f)$ (such as in (9.74a)) for the measurements. Much of the current literature on image restoration uses these types of unifying approaches. The book however considers more *ad hoc* methods.

9.5.1 Transformation to additive signal-independent noise

In some applications it is possible to find a (memoryless) operator that we can apply to the data $g[n,m]$ to transform the problem into one where the noise is additive and signal independent.

Example. The canonical example is speckle noise that arises when using coherent illumination (lasers, ultrasound, etc.), which is approximately multiplicative:

$$g[n,m] = f[n,m]\, v[n,m],$$

where $v[n,m]$ is random noise that is independent of $f[n,m]$.

What transformation would change this into an additive model? **??**

$$\log g[n,m] = \log f[n,m] + \log v[n,m]\,.$$

Procedure:

- Apply transformation (logarithm in this case)
- Apply noise-reduction method to the transformed data (*e.g.*, adaptive Wiener filter)
- Invert transformation (exponential in this case)

Example. Decorrelation of signal-dependent quantization noise in image coding (later).

9.5.2 Reduction of signal-dependent noise in the signal domain

Derives an approximate adaptive Wiener filter directly for the case of multiplicative noise, without taking the logarithm.

9.6

Temporal filtering for image restoration

The methods described in the preceding sections are all designed for the case where a single degraded image of some scene is available.

In applications such as remote sensing, forensics (from surveillance videos), and real-time medical imaging (*e.g.*, ultrasound), it is possible for there to be multiple images of the "same" scene. (If the scene or the image recorder are moving though...)

9.6.1

Frame averaging

If the scene and recorder are stationary, then the following model may apply:

$$g_k[n,m] = f[n,m] + v_k[n,m], \quad k = 1,\ldots,K.$$

Example. laser scanners for BW and color documents, mult-scan radar.

The natural approach to combining multiple frames is simply to take the arithmetic **average**:

$$\hat{f}[n,m] = \frac{1}{K}\sum_{k=1}^{K} g_k[n,m]\,.$$

If the noise is zero-mean and Gaussian, then this is the maximum likelihood (ML) estimator and also the minimum variance unbiased estimator of $f[n,m]$. For other noise models it may be appropriate to use other combinations (such as the median rather than the mean for Laplacian noise).

- As long as the above model holds (*i.e.*, no motion), the frame averaging can reduce noise without any loss of spatial resolution!
- For additional noise reduction, of course one can also smooth spatially (*e.g.*, if models for noise and signal PSD are unknown).

9.6.2

Motion-compensated image restoration

If the scene changes from one frame to the next, what will simple time-averaging do? **??**

To reduce such motion-induced artifacts, one can first estimate motion parameters (displacements) between image frames. This is called **registration**. One can then average or filter along motion trajectories.

Example. Using K frames, a natural estimator is:

$$\hat{f}[n,m,t_0] = \frac{1}{K}\sum_{k=1}^{K} g_k\left[n - d_{\mathrm{X}}^{(k)}, m - d_{\mathrm{Y}}^{(k)}\right],$$

where $(d_{\mathrm{X}}^{(k)}, d_{\mathrm{Y}}^{(k)})$ denotes the estimated displacement vector for the kth frame. (Typically one will need interpolation since the estimated displacements may be non-integer values.)

This method is appropriate under the following measurement model:

$$g_k[n,m] = f[n - v_{\mathrm{X}} \cdot (t_k - t_0), m - v_{\mathrm{Y}} \cdot (t_k - t_0), t_0]\,.$$

That model ignores within-frame motion blur, the importance of which depends on the frame rate.

9.7

Color images

Simple approach: represent color using RGB (or an alternative), apply (monochrome) restoration method to each of the three color planes, then recombine.

This approach disregards the correlation between color planes. Alternative approaches simultaneously process all color planes, *e.g.*, [9].

Summary

This chapter described methods for reducing noise, for reducing blurring, and for the combination of the two. Image restoration remains an active research area, particularly with regards to nonlinear algorithms. Most of the literature focuses either on the denoising problem alone (such as wavelet thresholding methods [6]) or considers *both* blur and noise simultaneously. Occasionally papers still appear that just consider the blur without treating the noise, *e.g.*, [10], but these are the exception since most researchers are fully aware of the instabilities of deconvolution and realize that one must simultaneously consider the effects of noise when trying to remove blur.

Bibliography

[1] H. C. Andrews and B. R. Hunt. *Digital image restoration*. Prentice-Hall, NJ, 1977.
[2] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. IoP, London, 1998.
[3] H. Stark. *Image recovery, theory and application*. Academic, Orlando, 1987.
[4] R. L. Lagendijk and J. Biemond. *Iterative identification and restoration of images*. Kluwer, Boston, 1991.
[5] R. H. T. Bates and M. J. McDonnell. *Image restoration and reconstruction*. Oxford, New York, 1986.
[6] D. L. Donoho. De-noising by soft-thresholding. *IEEE Tr. Info. Theory*, 41(3):613–27, May 1995.
[7] A. Antoniadis and J. Fan. Regularization and wavelet approximations. *J. Am. Stat. Ass.*, 96(455):939–55, September 2001.
[8] H. Stark and J. W. Woods. *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Englewood Cliffs, NJ, 1986.
[9] R. R. Schultz and R. L. Stevenson. Stochastic modeling and estimation of multispectral image data. *IEEE Tr. Im. Proc.*, 4(8):1109–19, August 1995.
[10] Y. Vardi and D. Lee. From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problems. *J. Royal Stat. Soc. Ser. B*, 55(3):569–612, 1993.