Notes on image coaddition, PSF homogenization, and matched filters

Eric H. Neilsen, Jr.
02 September 2009

CONTENTS

1 Introduction 1

2 A Matched Filter on a single exposure 1
2.1 The special case of an isolated point source in given location 1
2.2 Example 2
2.3 The Matched filter in Fourier space 3
2.4 A Fuzzy-minded explanation 4
2.5 An example 5

3 Application of matched filters to resampled images 6

4 Consequences of using a different filter for matching 7

5 PSF Homogenization 8

6 Naive matching filtering of a PSF homogenized image 10

7 Coaddition of PSF homogenized images 11
7.1 Coaddition in frequency space 11
7.2 Direct coaddition 11
7.3 Coaddition after homogenization 11
7.4 Coaddition after matched filtering 12

8 A Matched Filter on coadded, PSF homogenized exposures 12

9 A Matched Filter on naively coadded exposures 12

10 The Wiener Filter 12

11 Detection limits expressed in terms of sky noise 13

1 INTRODUCTION

As seen from the Earth, stars appear as point sources; the diameter of the star is usually too small to be measured. On images from telescopes, however, they appear slightly blurry due to turbulence in the atmosphere and (in space or in small telescopes) limitations in the optics of the instrument. The apparent morphology of a point source such as a star in an image called the point spread function (PSF), or sometimes the point response function, of the image. The PSF of different exposures from the same telescope may vary significantly, and the PSF can also vary across a single image.

In the approximation that the PSF is uniform across an image, the effect of the atmosphere and instrument on the image is that of the application of a low-pass Fourier filter.
There is a significant body of work dedicated to altering the point spread function of images, particularly in approximating the true sky from available images. Much of this work has been done specifically to address the early optical problems of the Hubble Space Telescope; see the STScI conference on restoration of HST images. Good reference papers include the review paper by Hunt and Digital Image Processing by Gonzales and Woods.

Sky surveys have an additional motivation for studying PSF alteration. When multiple exposures are combined into a single image of the sky, and the exposures do not all have the same PSF, the PSF in the combined image will jump discontinuously at the edge of each exposure. In a large survey with many exposures at differing offsets, this is a significant problem.

These notes were prepared for a meeting to discuss the impact of PSF homogenization in such a survey. The plots and examples below were all generated using the R statistical computing software package, and the R code accompanies the plots.

2 A Matched Filter on a Single Exposure

2.1 The special case of an isolated point source in given location

For an image of a single point source, the total number of counts recorded in each pixel of an image is:

\[ b(x) = f h(x, x_0) \]

Note that \( h \) can be a point spread function (scaled by the photometric solution), but could apply to more arbitrary mappings.

The final image will also contain noise:

\[ g(x) = b(x) + n(x) = f h(x, x_0) + n(x) \]

When we are concerned with objects near the limits of detection, the photon noise from the sky will dominate; the noise will be uniform over the image.

If we wanted to measure the flux from the point source, we might simply estimate it from the brightest pixel:

\[ \hat{f}_{\text{max}} = \frac{g(x_{\text{max}})}{h(x_{\text{max}}, x_0)} \]

We are not limited to using just that single pixel; we can get such an estimate for any pixel with flux:

\[ \hat{f}_x = \frac{g(x)}{h(x, x_0)} \]

but the uncertainty will not be uniform:

\[ \sigma_{\hat{f}_x} = \frac{\sigma_n}{h(x, x_0)} \]

We can do a weighted mean of these different estimates to get an estimate with minimal uncertainty:

\[ \hat{f} = \frac{\sum_x \frac{1}{\sigma_f^2} g(x)}{\sum_x \frac{1}{\sigma_f^2}} = \frac{\sum_x \frac{h^2(x, x_0)}{\sigma_f^2} \frac{g(x)}{h(x, x_0)}}{\sum_x \frac{h^2(x, x_0)}{\sigma_f^2}} = \frac{1}{\sum_x h^2(x, x_0)} \sum_x h(x, x_0) g(x) \]

Note that this is the value of the convolution of the image by its point spread function, evaluated at the position of the point source, and scaled by constant that depends on the point spread function and photometric solution:

\[ \hat{f} = \frac{\sum_x \left( \frac{h(x, x_0)}{\sum_x h^2(x, x_0)} \right) g(x, x_0)}{\sum_x m(x, x_0) g(x, x_0)} \]

Because the noise variance is independent of position, the uncertainty in the estimate of the flux is the standard deviation of the sky noise in the image, after convolution by the point spread function. Because we know this particular weighting minimizes the uncertainty in the flux relative to the flux, we can conclude that this weighting maximizes the value of the flux relative to the sky noise. This is precisely what we want for object detection; the point spread function is our optimal matching filter.
2.2 Example

First, I define a Moffat PSF:

```r
moffat <- function(r, fwhm, beta=4.765) {
  hwhm <- (fwhm/2.0)
  alpha <- hwhm*(2^((1.0/beta)-1)^(-0.5))
  p <- (1.0 + (r/alpha)^2.0)^(-1.0*beta)
  p/sum(p)
}
```

Plot the raw signal, with the sky sigma equal to the square root of the sky and the flux in a Moffat profile with a total sky flux 5 times the sky sigma:

```r
set.seed(53727)
npoints <- 64
sigma <- 4.0
flux <- 5*sigma
x <- (-1*npoints/2):(npoints/2)
h <- moffat(x,4.25)
n <- rnorm(x,sd=sigma)
b <- sigma^2+flux*h
g <- b+n

png("sample.png",height=256,width=640)
par(mar=c(1.1,1.1,0.1,8),bty="l",xaxt="n",yaxt="n",mgp=c(0,0,0))
plot(x,g,type="s",ylab="flux")
mtext("Data with noise",4,0,las=1)
lines(x,b,col="red",type="s")
mtext("Data w/o noise",4,0,las=1,padj=2,col="red")
dev.off()
```

Our representation of the pixel transfer function is centered on the center of the array rather than the edge, so the default R convolution shifts everything. Define a new convolution to center it on the 0 index.

```r
recenter <- function(x) {
  npts <- length(x)
  hnpts <- ceiling(npts/2)
  z <- c(x[(hnpts+1):npts],x[1:(hnpts)])
}
```

Now, apply the convolution

```r
fhat <- recenter(convolve(g,h,type="circular"))
```

```r
png("matched.png",height=256,width=640)
```
2.3 The Matched filter in Fourier space

Frequency filtering alters both the signal and noise of an image, but because the power spectrum of the signal and noise differ, the filtering will alter the signal and noise differently. To detect an image, we need the signal from a given object to be as high as possible relative to the noise; in other words, we need to apply a filter that maximizes the signal to noise of the objects near the detection limit.

Near the detection limit, the noise will be approximately uniform, and the power spectrum of the noise will be flat; the Fourier transform of the noise will be of equal magnitude at all frequencies, and have a mean of zero. The Fourier transform of the signal, however, will be the Fourier transform of the object, multiplied by the Fourier transform of the PSF. In the case of a point source, the power spectrum will be the shape of the PSF, with a magnitude that varies with the flux of the object.

So, our goal is to derive a matched filter that maximizes the signal to noise. In other words, our filtered image:

$$\hat{f}(x) = \sum_{x'} f(x') m(x - x')$$

should be a least squares fit, minimizing:

$$\chi^2 = \sum_x (f(x) - \hat{f}(x))^2$$

This is the Wiener filter, taking considering only low signal to noise objects as part of the signal. In Fourier space

$$M(\nu) = \frac{1}{H(\nu)} \frac{1}{1 + \frac{|N(\nu)|^2}{|B(\nu)|^2}}$$

$$= \frac{|F(\nu)|^2 H^*(\nu)}{|B(\nu)|^2 + |N(\nu)|^2}$$

If we are looking at low signal to noise objects ($|B|^2 << |N|^2$) and we are looking at a random distribution of point sources ($F(\nu)$ is flat), then

$$M(\nu) \propto H^*(\nu)$$

and the matched filter is a mirror reflection of the point spread function, normalized by some factor (which doesn’t affect the effectiveness of the matched filter).
2.4 *A Fuzzy-minded explanation*

What, then, does this mean? A pixel in the "true" image $f(x)$ has contributions from each $F(\nu)$ Fourier components. Convolution by the point spread function by the instrument, atmosphere, etc. reduces this amount to $H(\nu)F(\nu)$, and noise $N(\nu)$ is added to the result, giving us our observed image $g(x)$, whose Fourier transform is:

$$G(\nu) = H(\nu)F(\nu) + N(\nu)$$

Our best estimate of $F(\nu)$ will be

$$\hat{F}(\nu) = \frac{G(\nu)}{H(\nu)}$$

Our image $g(x)$, however, does not have the same signal to noise in each frequency component. If we weight each frequency component by its $(S/N)^2$ rather than doing a direct inverse transform, then we get

$$\hat{f}(x) \propto \sum_\nu \frac{|F(\nu)H(\nu)|^2}{|N(\nu)|^2} \frac{G(\nu)}{H(\nu)} e^{\frac{2\pi i x}{N}}$$

which is just the inverse transform of our match filtered image.

2.5 *An example*

Going back to our previous example,

```r
GG <- abs(fft(g))^2
BB <- abs(fft(b))^2
NN <- abs(fft(n))^2
FFhat <- abs(fft(fhat))^2
k <- 1:(npoints/2)
```

```r
gg <- abs(fft(g))^2
bb <- abs(fft(b))^2
mn <- abs(fft(n))^2
ffhat <- abs(fft(fhat))^2
k <- 1:(npoints/2)
```

```r
png("matchedps.png",height=256,width=640)
par(mar=c(4,4,0.1,0.1),bty="n")
plot(k,GG[k],type="s",log="y",lwd=3,xlab="wavenumber",ylab="power")
lines(k,FFhat[k],type="s",col="blue")
lines(k,BB[k],type="s",col="green")
lines(k,NN[k],type="s",col="red")
leglabels<-c("Original data","Matched filtered data","Noise-free data","Noise")
legcols<-c("black","blue","green","red")
legend(20,1e+06,leglabels,col=legcols,lty=1,bty="n")
dev.off()
```

Of course, when we are doing our fit, we do not know specifically what all the noise values are, so we use our expectation value for our magnitude of the noise:

```
```
NNe <- rep(0, times=npoints+1) + npoints * (sigma^2)
GGe <- rep(0, times=npoints+1) + BB + NNe

png("matchedpsue.png", height=256, width=640)
par(mar=c(4,4,0.1,0.1), bty="l")
plot(k, GG[k], type="s", log="y", lwd=2, col="gray", xlab="wavenumber", ylab="power")
lines(k, FFhat[k], type="s", col="blue")
lines(k, GGe[k], type="s", lwd=2)
lines(k, BB[k], type="s", col="green")
lines(k, NNe[k], type="s", col="red")
leglabels<-c("Original data","Matched filtered data","Expected signal+noise", "Noise-free signal")
legcols<-c("gray","blue","black","green","red")
legend(20,1e+06,leglabels, col=legcols, lty=1, bty="n")
dev.off()

Interpolation by a sinc function that is critically sampled at a higher frequency than the point spread function, such that it doesn’t smooth the original PSF in physical space, is benign with respect to the matched filter; the optimal matched filter for point sources is still the PSF (but in the image space of the interpolated image).

Recall that the Fourier transform of a sinc function is a square; interpolation by a sinc is equivalent to taking the Fourier transform, possibly replacing frequencies higher that some threshold with zero, padding the result with zeros to get an array of the desired size, and inverting the Fourier transform. For example, in the above example, the power spectro of the intrepolated images would look like this:

3 Application of matched filters to resampled images

Interpolation by a sinc function that is critically sampled at a higher frequency than the point spread function, such that it doesn’t smooth the original PSF in physical space, is benign with respect to the matched filter; the optimal matched filter for point sources is still the PSF (but in the image space of the interpolated image).

Recall that the Fourier transform of a sinc function is a square; interpolation by a sinc is equivalent to taking the Fourier transform, possibly replacing frequencies higher that some threshold with zero, padding the result with zeros to get an array of the desired size, and inverting the Fourier transform. For example, in the above example, the power spectro of the intrepolated images would look like this:
The works because, although there is significant correlation in the noise in the interpolated image, if a sinc function with a reasonable width was used, all of the correlation is in frequencies where the matched filter has no power, and therefore these frequencies have zero weight in the filtered image.

Spline interpolation gives nearly the same answer as sinc interpolation, so it is nearly benign as well:

```r
spg <- spline(g,n=(snpoints+1))$y
sph <- spline(h,n=(snpoints+1))$y
spfhat <- recenter(convolve(spg,sph,type="circular"))
spGG <- abs(fft(spg))^2
spFFhat <- abs(fft(spfhat))^2
```

Note that, while the high frequency power in the spline interpolated version does not go precisely to zero as it does in the sinc interpolated version, the contribution from these frequencies is still many orders of magnitude less than in lower frequencies.
4 Consequences of Using a Different Filter for Matching

Consider the situation where, instead of using a filter that matches our PSF, we use a different filter, $H_0$, for object detection, perhaps one that matches the global average PSF of many observations.

Instead of using an optimal image for object detection:

$$\hat{F}(\nu) = H^*(\nu)G(\nu)$$

we get a slightly different image:

$$\hat{F}_0(\nu) = H_0^*(\nu)G(\nu)$$
$$= \frac{H_0^*(\nu)}{H^*(\nu)} \hat{F}(\nu)$$

Because the $H_0$ is probably close to $H$, this is likely to be fairly benign. However, recall one important assumption we made in deriving this matched filter: that the image was a random distribution of point sources. Different matched filters will match objects with different morphologies. If, for example, $H_0$ is slightly wider than $H$, the filter may not be optimally suited for detecting point sources, but it may be better suited for detecting slightly extended sources. If $H_0$ has a different ellipticity than $H$, then it will be better at detecting objects with some ellipticities, and worse at detecting objects with others. So, if we use the same matched filter for all images, ignoring variations is PSF, then our limiting magnitude will depend on object morphology in some complicated way.

5 PSF Homogenization

PSF homogenization is the process of convolving an image by some kernel that results in an image with a desired PSF. This is easiest to understand in Fourier space. Consider a starting image with PSF $H$:

$$G(\nu) = H(\nu)F(\nu) + N(\nu)$$

If we want an image $G_0(\nu)$ such that

$$G_0(\nu) = H_0(\nu)F(\nu) + N_0(\nu)$$

then we must multiply $G(\nu)$ by

$$\frac{H_0(\nu)}{H(\nu)}$$

to get

$$G_0(\nu) = \frac{H_0(\nu)}{H(\nu)}G(\nu)$$
$$= \frac{H_0(\nu)}{H(\nu)}H(\nu)F(\nu) + \frac{H_0(\nu)}{H(\nu)}N(\nu)$$
$$= H_0(\nu)F(\nu) + \frac{H_0(\nu)}{H(\nu)}N(\nu)$$
$$= H_0(\nu)F(\nu) + N_0(\nu)$$

where

$$N_0(\nu) = \frac{H_0(\nu)}{H(\nu)}N(\nu)$$

Note that the point spread function is in the denominator of the homogenization filter: the filter amplifies spatial frequencies where there is little power in the actual PSF relative to the desired PSF, while it suppresses frequencies where there is more power.

Consider the example we have been using:
GG <- abs(fft(g))^2
BB <- abs(fft(b))^2
NN <- abs(fft(n))^2
NNe <- rep(0, times=npoints+1) + npoints * (sigma^2)
k <- 1:(npoints/2)

png("hmsampstart.png", height=256, width=640)
par(mar=c(4,4,0.1,0.1), bty="l")
plot(k, GG[k], type="s", log="y", lwd=2, xlab="wavenumber", ylab="power", col="grey")
lines(k, BB[k], type="s", col="lightgreen")
lines(k, NNe[k], type="s", col="pink")
leglabels<-c("Original data", "Noise-free data", "Expected noise")
legcols<-c("gray", "lightgreen", "pink")
legend(20, 1e+06, leglabels, col=legcols, lty=1, bty="n")
dev.off()

Now, let us construct and apply a homogenization filter. Start with the case where the target PSF is wider (has weaker high frequency spatial components):

h0 <- moffat(x, 6.0)
GGh <- abs(fft(h0) * fft(g) / fft(h))^2
NNh <- NNe * abs(fft(h0) / fft(h))^2
BBh <- abs(fft(h0) * fft(b) / fft(h))^2

png("hmsampwide.png", height=256, width=640)
par(mar=c(4,4,0.1,0.1), bty="l")
plot(k, GG[k], type="s", log="y", lwd=2, xlab="wavenumber", ylab="power", col="grey")
lines(k, GGh[k], type="s", lwd=2)
lines(k, BBh[k], type="s", col="green")
lines(k, NNh[k], type="s", col="red")
leglabels<-c("Original data", "Homogenized data", "Homogenized signal", "Homogenized expected noise")
legcols<-c("gray", "black", "green", "red")
legend(15, 1e+06, leglabels, col=legcols, lty=1, bty="n")
dev.off()

Now we move to the case where the target PSF is sharper (has stronger high frequency spatial components):
\[ h_0 \leftarrow \text{moffat}(x, 2.5) \]
\[ \text{GGh} \leftarrow |\text{fft}(h_0) \ast \text{fft}(g)/\text{fft}(h)|^2 \]
\[ \text{NNh} \leftarrow |NNe \ast \text{fft}(h_0)/\text{fft}(h)|^2 \]
\[ \text{BBh} \leftarrow |\text{fft}(h_0) \ast \text{fft}(b)/\text{fft}(h)|^2 \]

\begin{verbatim}
png("hmsampsharp.png", height=256, width=640)
par(mar=c(4,4,0.1,0.1), bty="l")
plot(k, GGh[k], type="s", log="y", lwd=2, xlab="wavenumber", ylab="power", col="grey")
lines(k, BBh[k], type="s", col="green")
lines(k, NNh[k], type="s", col="red")
leglabels<-c("Original data","Homogenized data","Homogenized signal", "Hmg. expected noise")
legcols<-c("gray","black","green","red")
legend(20,1e+06,leglabels,col=legcols,lty=1,bty="n")
dev.off()
\end{verbatim}

Note that the high frequency noise gets greatly amplified. This effect can be reduced by a more careful choice for the form for the target PSF (for example by using a sinc instead of a gaussian), but to at least some extent amplification of high frequency noise is inherent in the technique.

6 Naive matching filtering of a PSF homogenized image

Our derivation of the matching filter in previous sections assumed that the noise was uniform, at least over the range for the frequency range has finite values. This is clearly not the case for the PSF homogenized image. What happens if we ignore this, and pretend that the noise is indeed uniform after homogenization? The final, matched filtered image would look like this:

\[ \hat{F}_h(\nu) = H_0^*(\nu)G_h(\nu) \]
\[ = \frac{|H_0(\nu)|^2}{H(\nu)}G(\nu) \]
\[ = \frac{|H_0(\nu)|^2}{|H(\nu)|^2}H^*(\nu)G(\nu) \]
\[ = \left( \frac{|H_0(\nu)|}{|H(\nu)|} \right)^2 \hat{F}(\nu) \]

The fractional difference between applying an optimal matched filter and naively filtering a homogenized image is the square of the difference between the optimal filter and simple application of a standardized filter on non-homogenized images. When using a standardized PSF, homogenization pushes the weighting further from optimal, and exaggerates the sensitivity of the detection limit to object morphology.

What, then, is the optimal filter for an homogenized image?

\[ \hat{F}(\nu) = H^*(\nu)G(\nu) \]
\[
\begin{align*}
\mathcal{F}(g_1(x) + g_2(x)) &= \mathcal{F}(g_1(x)) + \mathcal{F}(g_2(x)) \\
f(x) + g(x) &= \mathcal{F}^{-1}[G_1(\nu) + G_2(\nu)]
\end{align*}
\]

### 7 Coaddition of PSF Homogenized Images

#### 7.1 Coaddition in Frequency Space

The effects of PSF homogenization are much clearer in frequency space than in image space, so in this discussion I will make use of the addition theorem (see Bracewell, The Fourier Transform and its Applications, p104):

\[
\begin{align*}
\mathcal{F}(g_1(x) + g_2(x)) &= \mathcal{F}(g_1(x)) + \mathcal{F}(g_2(x)) \\
f(x) + g(x) &= \mathcal{F}^{-1}[G_1(\nu) + G_2(\nu)]
\end{align*}
\]

#### 7.2 Direct Coaddition

What happens if we coadd images? If we use the traditional approach, adding the images:

\[
\begin{align*}
G_1(\nu) &= H_1(\nu)F(\nu) + N_1(\nu) \\
G_2(\nu) &= H_2(\nu)F(\nu) + N_2(\nu)
\end{align*}
\]

with weights \(w_1\) and \(w_2\), perhaps setting \(w_2 = 1 - w_1\), then we get

\[
G_a(\nu) = [w_1H_1(\nu) + w_2H_2(\nu)]F(\nu) + w_1N_1(\nu) + w_2N_2(\nu)
\]

#### 7.3 Coaddition after Homogenization

After homogenization, we have:

\[
\begin{align*}
G_{1h}(\nu) &= \frac{H_0(\nu)}{H_1(\nu)}G_1(\nu) \\
&= H_0(\nu)F(\nu) + \frac{H_0(\nu)}{H_1(\nu)}N_1(\nu)
\end{align*}
\]

\[
\begin{align*}
G_{2h}(\nu) &= \frac{H_0(\nu)}{H_2(\nu)}G_2(\nu) \\
&= H_0(\nu)F(\nu) + \frac{H_0(\nu)}{H_2(\nu)}N_2(\nu)
\end{align*}
\]

and after coaddition:

\[
G_{ah}(\nu) = w_1\frac{H_0(\nu)}{H_1(\nu)}G_1(\nu) + w_2\frac{H_0(\nu)}{H_2(\nu)}G_2(\nu) = H_0(\nu)F(\nu) + \frac{H_0(\nu)}{H_1(\nu)}w_1N_1(\nu) + \frac{H_0(\nu)}{H_2(\nu)}w_2N_2(\nu)
\]

Note that for any given \(\nu\), the coadded \(G_{ah}(\nu)\) is still a weighted average of \(G_1(\nu)\) and \(G_2(\nu)\), but this time the weights are a function of \(\nu\). Note that the signal to noise \(G_1(\nu)\) at a given \(\nu\) is proportional to \(H_1(\nu)\), but that \(H_1(\nu)\) appears in the denominator of the varying weights: the weight of a given pixel in frequency space is inversely proportional to its signal to noise! At frequencies where \(G_1\) has a high signal to noise but \(G_2\) has a low one, the \(G_2\) will dominate in the coadded image, and vice versa. This is exactly the opposite of the weighting one wants to maximize the signal to noise in the coadded image.

If you examine the weights of \(N_1(\nu)\) and \(N_2(\nu)\), you can see that the weights of either can become arbitrarily large, depending on the corresponding \(H\). Note that this is not possible in direct coaddition, where the weight is independent of frequency.
7.4 Coaddition after matched filtering

After application of the matched filter, we have

\[ G_{1m}(\nu) = H_1^*(\nu)G_1(\nu) = |H_1(\nu)|^2 F(\nu) + H_1^*(\nu)N_1(\nu) \]

\[ G_{2m}(\nu) = H_2^*(\nu)G_2(\nu) = |H_2(\nu)|^2 F(\nu) + H_2^*(\nu)N_2(\nu) \]

and after coaddition:

\[ G_{am}(\nu) = w_1H_1^*(\nu)G_1(\nu) + w_2H_2^*(\nu)G_2(\nu) = [w_1|H_1(\nu)|^2 + w_2|H_2(\nu)|^2] F(\nu) + w_1H_1^*(\nu)N_1(\nu) + w_2H_2^*(\nu)N_2(\nu) \]

Now, the relative weights of \( G_1 \) and \( G_2 \) still vary with \( \nu \), but in a more appropriate way: frequencies with more signal get weighted more heavily.

At frequencies where a given exposure has a small value for \( H(\nu) \), note that the noise contribution to the coadded image is suppressed.

8 A Matched Filter on coadded, PSF homogenized exposures

Going back to the step in the matched filter derivation before we assume uniform noise, and substituting in the noise from above, we get:

\[ M(\nu) = \frac{H_0^*(\nu)}{|H_0(\nu)|^2 w_1N_1(\nu) + w_2N_2(\nu)|^2} \]

Note that, for an \( \nu \) for which \( H(\nu) \ll H_0(\nu) \), that term is likely to dominate the denominator, reducing the weight for the frequency. For coadded homogenized images, the image with the worst seeing determines which frequencies can be used for detection.

If we make our target PSF such that it is wider than the input images, this will not be a problem. Instead, the numerator will drop high frequency components to small values.

9 A Matched Filter on naively coadded exposures

Traditionally, astronomers first align and add multiple exposures, and then apply a matched filter. Because the interpolation can be done such that the noise remains flat over the frequency range required, the simple use of the measured PSF as the matching filter on such an image is justified.

This approach may, however, may not be taking maximum advantage of the available data.

10 The Wiener Filter

Using Parseval’s theorem (see Bracewell, The Fourier Transform and its Applications, p112):

\[ \sum_x |f(x)|^2 = \frac{1}{N} \sum_\nu |F(\nu)|^2 \]

and the addition theorem (see Bracewell, The Fourier Transform and its Applications, p104):

\[ \mathcal{F}(f(x) + g(x)) = \mathcal{F}(f(x)) + \mathcal{F}(g(x)) \]

we know we can do the minimization in Fourier space:

\[ \chi^2 = \frac{1}{N} \sum_\nu |F(\nu) - \hat{F}(\nu)|^2 \]
where the convolution by the filter can be expressed more conveniently:

\[
\hat{F}(\nu) = M(\nu)G(\nu) = M(\nu)[H(\nu)F(\nu) + N(\nu)] = M(\nu)H(\nu)F(\nu) + M(\nu)N(\nu)
\]

We can now minimize:

\[
\chi^2 \propto \sum_{\nu} |F(\nu) - M(\nu)H(\nu)F(\nu) - M(\nu)N(\nu)|^2
\]

We then need to differentiate the above equation with respect to M, and set the derivative to 0 to find the minimum. If we assume the signal and noise are not correlated, this ultimately reduces to

\[
M(\nu) = \frac{|F(\nu)|^2 H^*(\nu)}{|B(\nu)|^2 + |N(\nu)|^2}
\]

If we are looking at low signal to noise objects,

\[
|B(\nu)|^2 \ll |N(\nu)|^2
\]

so

\[
M(\nu) \propto |F(\nu)|^2 H^*(\nu)
\]

If the noise is uniform,

\[
M(\nu) \propto |F(\nu)|^2 H^*(\nu)
\]

and if the signal is a uniform distribution of point sources,

\[
M(\nu) \propto H^*(\nu)
\]

11 Detection limits expressed in terms of sky noise

After convolution by a matching filter:

\[
\hat{f}(x) = \sum_{x_i} m(x, x_i) g(x_i)
\]

the variance of uniform noisy sky will be

\[
\sigma_n^2(x) = \sum_{x_i} m(x, x_i) \sigma_n^2
\]

so the noise in the sky drops by a factor of the uniformly weighted mean of the squares of the filter values.

The peak of an object that matches the filter can be thought of as dropping weighted mean of these values, where filtering pixels with larger values get weighted more heavily; the peak drops by less than the noise, making the same object appear more standard deviations away from the noise.

The specification for detection depth as a function of “sky sigma” is therefore meaningless for comparing matching filters. Detection at a given sigma threshold is possible for any matching filter, but different filters will significantly affect the physical flux that corresponds to.

What a specification of depths at a given sky sigma does restrict is the number of false detections.