THE UNIVERSITY *of* TEXAS

Convolution, Noise and Filters

Philip Baldwin, Ph.D.
Department of Biochemistry

Response to an Entire Signal

The response of a system with impulse response $h(t)$ to input $x(t)$ is simply the convolution of $x(t)$ and $h(t)$:

$$x(t) \to y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

One Way to Think of Convolution

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau$$

$$x[j] * h[j] = \sum_{k} x[k] \cdot h[j - k]$$

Think of it this way:

- Shift a copy of *h* to each position *t* (or discrete position *k*)
- Multiply by the value at that position $x(t)$ (or discrete sample $x[k]$)
- Add shifted, multiplied copies for all *t* (or discrete *k*)

Example: Convolution

$$x[j] = [\quad 1 \quad 4 \quad 3 \quad 1 \quad 2 \quad]$$
$$h[j] = [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad]$$

$$x[0]\, h[j-0] = [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$
$$x[1]\, h[j-1] = [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$
$$x[2]\, h[j-2] = [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$
$$x[3]\, h[j-3] = [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$
$$x[4]\, h[j-4] = [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$

$$x[j] * h[j] = \sum_k x[k]\, h[j-k]$$
$$= [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]$$

Example: Convolution

$$
\begin{array}{rcl}
x[j] & = & [\;\; 1 \quad 4 \quad 3 \quad 1 \quad 2 \;\;] \\
h[j] & = & [\;\; 1 \quad 2 \quad 3 \quad 4 \quad 5 \;\;]
\end{array}
$$

$$
\begin{array}{rcl}
x[0]\, h[j-0] & = & [\;\; 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[1]\, h[j-1] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[2]\, h[j-2] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[3]\, h[j-3] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[4]\, h[j-4] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;]
\end{array}
$$

$$
\begin{array}{rcl}
x[j] * h[j] & = & \displaystyle\sum_k x[k]\, h[j-k] \\
 & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;]
\end{array}
$$

Example: Convolution

$$
\begin{array}{rcl}
x[j] & = & [\;\; 1 \quad 4 \quad 3 \quad 1 \quad 2 \;\;] \\
h[j] & = & [\;\; 1 \quad 2 \quad 3 \quad 4 \quad 5 \;\;]
\end{array}
$$

$$
\begin{array}{rcl}
x[0]\,h[j-0] & = & [\;\; 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[1]\,h[j-1] & = & [\;\; \_\_ \quad 4 \quad 8 \quad 12 \quad 16 \quad 20 \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[2]\,h[j-2] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[3]\,h[j-3] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;] \\
x[4]\,h[j-4] & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;]
\end{array}
$$

$$
\begin{array}{rcl}
x[j] * h[j] & = & \sum_k x[k]\,h[j-k] \\
 & = & [\;\; \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \;\;]
\end{array}
$$

Example: Convolution

$$
\begin{array}{rcl}
x[j] & = & [\quad 1 \quad 4 \quad 3 \quad 1 \quad 2 \quad] \\
h[j] & = & [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad]
\end{array}
$$

$$
\begin{array}{rcl}
x[0]\, h[j-0] & = & [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad] \\
x[1]\, h[j-1] & = & [\quad \_\_ \quad 4 \quad 8 \quad 12 \quad 16 \quad 20 \quad \_\_ \quad \_\_ \quad \_\_ \quad] \\
x[2]\, h[j-2] & = & [\quad \_\_ \quad \_\_ \quad 3 \quad 6 \quad 9 \quad 12 \quad 15 \quad \_\_ \quad \_\_ \quad] \\
x[3]\, h[j-3] & = & [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad] \\
x[4]\, h[j-4] & = & [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]
\end{array}
$$

$$
\begin{array}{rcl}
x[j] * h[j] & = & \sum_k x[k]\, h[j-k] \\
 & = & [\quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad]
\end{array}
$$

Example: Convolution

$$
\begin{aligned}
x[j] &= [\quad 1 \quad 4 \quad 3 \quad 1 \quad 2 \quad] \\
h[j] &= [\quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad]
\end{aligned}
$$

$$
\begin{array}{rcl}
x[0]\,h[j-0] & = & [\; 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \;] \\
x[1]\,h[j-1] & = & [\; \underline{\quad} \quad 4 \quad 8 \quad 12 \quad 16 \quad 20 \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \;] \\
x[2]\,h[j-2] & = & [\; \underline{\quad} \quad \underline{\quad} \quad 3 \quad 6 \quad 9 \quad 12 \quad 15 \quad \underline{\quad} \quad \underline{\quad} \;] \\
x[3]\,h[j-3] & = & [\; \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \underline{\quad} \;] \\
x[4]\,h[j-4] & = & [\; \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \;]
\end{array}
$$

$$
\begin{aligned}
x[j] * h[j] &= \sum_k x[k]\, h[j-k] \\
&= [\; \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \quad \underline{\quad} \;]
\end{aligned}
$$

Example: Convolution

$$
\begin{array}{rcl}
x[j] & = & [\ 1\quad 4\quad 3\quad 1\quad 2\ ] \\
h[j] & = & [\ 1\quad 2\quad 3\quad 4\quad 5\ ]
\end{array}
$$

$$
\begin{array}{rcccccccccccc}
x[0]\,h[j-0] & = & [& 1 & 2 & 3 & 4 & 5 & \_\_ & \_\_ & \_\_ & \_\_ &] \\
x[1]\,h[j-1] & = & [& \_\_ & 4 & 8 & 12 & 16 & 20 & \_\_ & \_\_ & \_\_ &] \\
x[2]\,h[j-2] & = & [& \_\_ & \_\_ & 3 & 6 & 9 & 12 & 15 & \_\_ & \_\_ &] \\
x[3]\,h[j-3] & = & [& \_\_ & \_\_ & \_\_ & 1 & 2 & 3 & 4 & 5 & \_\_ &] \\
x[4]\,h[j-4] & = & [& \_\_ & \_\_ & \_\_ & \_\_ & 2 & 4 & 6 & 8 & 10 &]
\end{array}
$$

$$
x[j] * h[j] = \sum_k x[k]\,h[j-k]
$$

$$
= [\ \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\quad \_\_\ ]
$$

Example: Convolution

$$x[j] = [\ 1 \quad 4 \quad 3 \quad 1 \quad 2\ ]$$

$$h[j] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5\ ]$$

$$x[0]\,h[j-0] = [\ 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_ \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[1]\,h[j-1] = [\ \_\_ \quad 4 \quad 8 \quad 12 \quad 16 \quad 20 \quad \_\_ \quad \_\_ \quad \_\_\ ]$$

$$x[2]\,h[j-2] = [\ \_\_ \quad \_\_ \quad 3 \quad 6 \quad 9 \quad 12 \quad 15 \quad \_\_ \quad \_\_\ ]$$

$$x[3]\,h[j-3] = [\ \_\_ \quad \_\_ \quad \_\_ \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad \_\_\ ]$$

$$x[4]\,h[j-4] = [\ \_\_ \quad \_\_ \quad \_\_ \quad \_\_ \quad 2 \quad 4 \quad 6 \quad 8 \quad 10\ ]$$

$$x[j] * h[j] = \sum_k x[k]\,h[j-k]$$

$$= [\ 1 \quad 6 \quad 14 \quad 23 \quad 34 \quad 39 \quad 25 \quad 13 \quad 10\ ]$$

Example: Two-Dimensional Convolution

$$
\begin{matrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{matrix}
\quad * \quad
\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{matrix}
\quad =
$$

$$
\begin{matrix}
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} \\
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} \\
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} \\
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} \\
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} \\
\underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad} & \underline{\qquad}
\end{matrix}
$$

Example: Two-Dimensional Convolution

$$
\begin{matrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{matrix}
\quad * \quad
\begin{matrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{matrix}
\quad =
$$

$$
\begin{matrix} 1 & 2 & 4 & 5 & 4 & 2 \\ 2 & 5 & 9 & 12 & 10 & 4 \\ 3 & 7 & 13 & 17 & 14 & 6 \\ 3 & 7 & 13 & 17 & 14 & 6 \\ 2 & 5 & 9 & 12 & 10 & 4 \\ 1 & 2 & 4 & 5 & 4 & 2 \end{matrix}
$$

Properties of Convolution

- Commutative: $f * g = g * f$
- Associative: $f * (g * h) = (f * g) * h$
- Distributive over addition: $f * (g + h) = f * g + f * h$
- Derivative: $\frac{d}{dt}(f * g) = f' * g + f * g'$

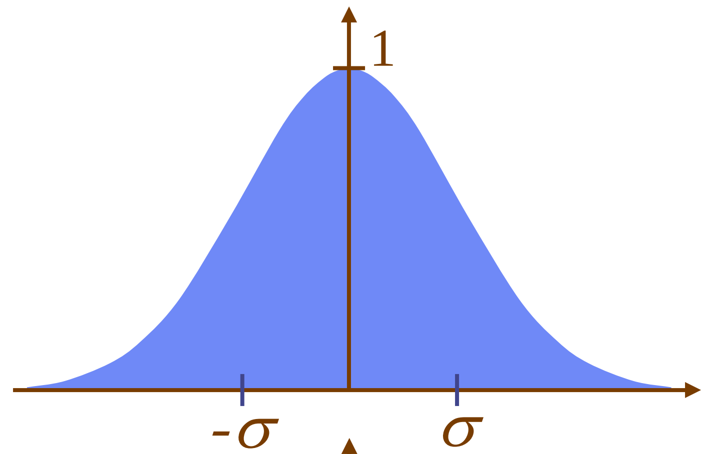
Convolution has the same mathematical properties as multiplication

(This is no coincidence, see Fourier convolution theorem!)

Gaussian

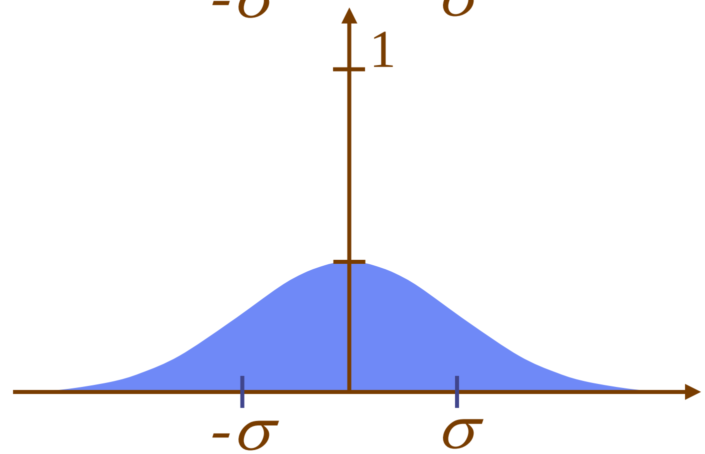
Gaussian: maximum value = 1

$$G(t,\sigma) = e^{-t^2/2\sigma^2}$$



Normalized Gaussian: area = 1

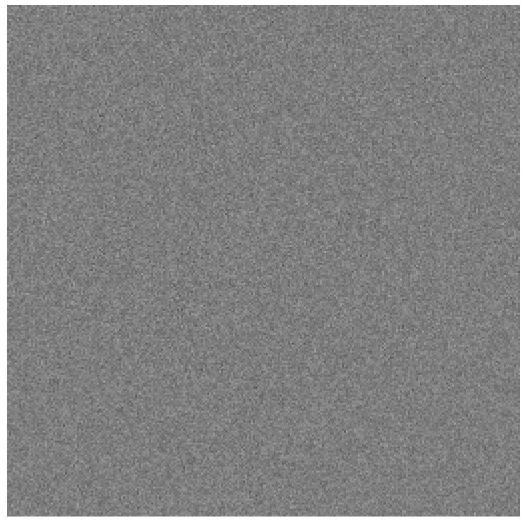
$$G(t,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-t^2/2\sigma^2}$$



Convolving a Gaussian with another:

$$G(t,\sigma_1) * G(t,\sigma_2) = G(t,\sqrt{\sigma_1^2 + \sigma_2^2})$$

What is Noise?


+

=

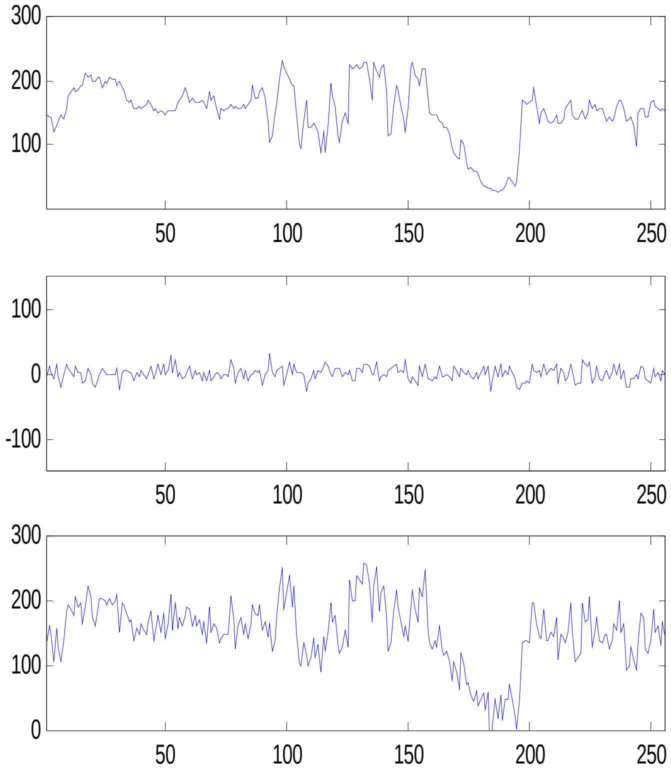

image

noise

‘grainy’
image

What is Noise?

- Anything that is NOT signal:
  - Signal is what carries information that we are interested in
  - Noise is anything else
- Noise may be
  - Completely random (both spatially and temporally)
  - Structured
  - Structured randomness



Statistical Review

Mean: The average or expected value

$$\mu = E\{x\} = \frac{1}{N}\sum x$$

Variance: The expected value of the squared error

$$\sigma^2 = E\{(x-\mu)^2\} = E\{x^2\} - \mu^2$$

Standard Deviation: The square root of the variance

$$\sigma = \sqrt{\sigma^2}$$

Ensembles of Images

Consider the picture $\tilde{I}(x)$ as a random variable from which we sample an ensemble of images from the space of all possibilities

This ensemble (or collection) of images has a mean (average) image, $\bar{I}(x)$

If we sample enough images, the ensemble mean approaches the noise-free original signal

- Often not feasible

Signal-To-Noise Ratio

If we compare the strength of a signal or image (the mean of the ensemble) to the variance between individual acquired images we get a signal-to-noise ratio:

$$SNR = \frac{\mu}{\sigma}$$

The better (higher) the SNR, the better our ability to discern the signal information

Problem: How to measure m to compute the SNR?

Noise and the Frequency Domain

Noisy input:

$$\tilde{I}(x) = \bar{I}(x) + \tilde{n}(x)$$

Spectrum of noisy input:

$$\mathcal{F}(\tilde{I}(x)) = \mathcal{F}(\bar{I}(x)) + \mathcal{F}(\tilde{n}(x))$$

- White noise has equally random amounts of all frequencies
- “Colored” noise has unequal amount for different frequencies
- Since signals often have more low frequencies than high, the effect of white noise is usually greatest for high frequencies

Filters

- Low pass filter
  - eliminate high frequencies and leave the low frequencies.
- High pass filter
  - eliminate low frequencies and leave high frequencies.
- Band pass filter
  - only a limited range of frequencies remains
- Gaussian smoothing
  - has the effect of cutting off the high frequency components of the frequency spectrum

Low-Pass Filter

- Recall that quick changes in a signal/image require high frequencies

- High frequency details are often “buried” in noise, which also requires high frequencies

- One method of reducing noise is pixel averaging:
  - Average same pixel over multiple images of same scene
  - Average multiple (neighboring) pixels in single image

Convolution Filtering: Averaging

Can use a square function ("box filter") or Gaussian to locally average the signal/image

- Square (box) function: uniform averaging
- Gaussian: center-weighted averaging

Both of these blur the signal or image

Low-Pass Filtering = Spatial Blurring

Low-pass filtering and spatial blurring are the same thing

Any convolution kernel with all positive (or all negative) weights does:

- Weighted averaging
- Spatial blurring
- Low-pass filtering

They are all equivalent

Filtering and Convolution

Two ways to think of general filtering:

- Spatial: Convolution by some spatial-domain kernel
- Frequency: Multiplication by some frequency-domain filter

Can implement/analyze either way

Low-Pass Filtering

Tradeoff:

Reduces Noise

but

Blurs Image

The worse the noise, the more you need to blur to remove it

Original

After Low-pass filtering

"Ideal" Low-Pass Filtering

For cutoff frequency u_c:

$$H(u) = \Pi(u/u_c) = \begin{cases} 1 & \text{if } |u| \le u_c \\ 0 & \text{otherwise} \end{cases}$$

What is the corresponding convolution kernel?

What problem does this cause?

What could you do differently?

Better (Smoother) Low-Pass Filtering

Gentler ways of cutting off high frequencies:

- Hanning

$$H(u) = \begin{cases} 0.5 + 0.5\cos(\frac{\pi}{2} u / u_c) & \text{if } |u| \le u_c \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian

$$H(u) = e^{-u^2 / 2u_c^2}$$

- Butterworth

$$H(u) = \frac{1}{1 + \left(u^2 / u_c^2 \right)^n}$$

n controls the sharpness of the cutoff

Sharpening

- Blurring is low-pass filtering, so de-blurring is high-pass filtering:
  - Explicit high-pass filtering
  - Unsharp Masking
  - Deconvolution
  - Edge Detection
- Tradeoff:
  - Reduces Blur

    but
  - Increases Noise

High-Pass Filtering

- "Ideal":

$$H(u) = 1 - \Pi(u/u_c) = \begin{cases} 0 & \text{if } |u| \le u_c \\ 1 & \text{otherwise} \end{cases}$$

- Flipped Butterworth:

$$H(u) = 1 - \frac{1}{1 + (u^2/u_c^2)^n}$$

High-Pass Filtering vs. Low-Pass Filtering

Original

After Low-pass filtering

After High-pass filtering

Convolution Filtering: Unsharp Masking

Unsharp masking is a technique for high-boost filtering. To sharpen a signal/image, subtract a little bit of the blurred input.

Procedure:

- Blur the image.
- Subtract from the original.
- Multiply by some weighting factor.
- Add back to the original.

$$I' = I + \alpha(I - I * g)$$

where I' is the original image, g is the smoothing (blurring) kernel, and I is the final (sharpened) image

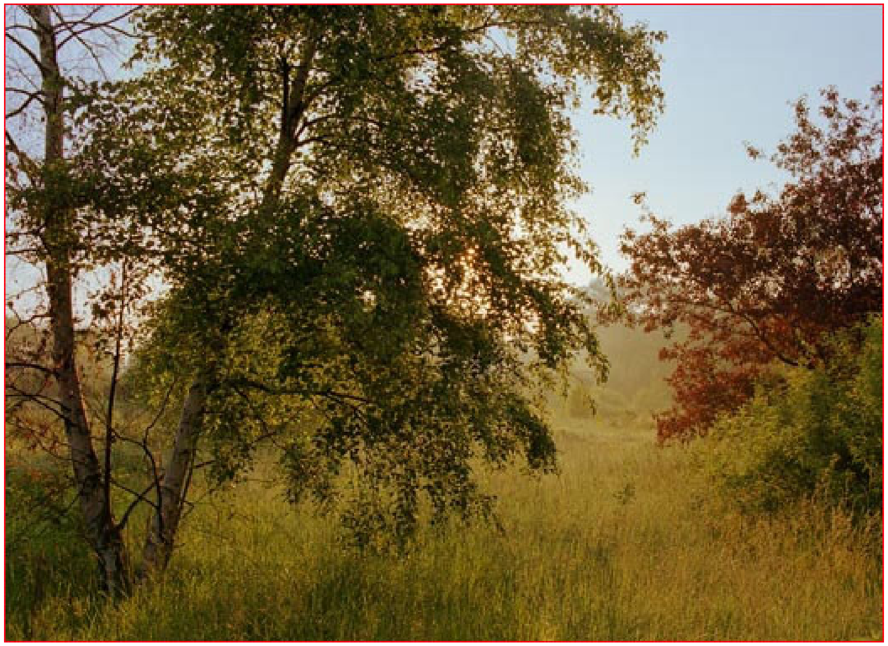
Unsharp Masking: Implementation

$$I + \alpha(I - I * g)$$

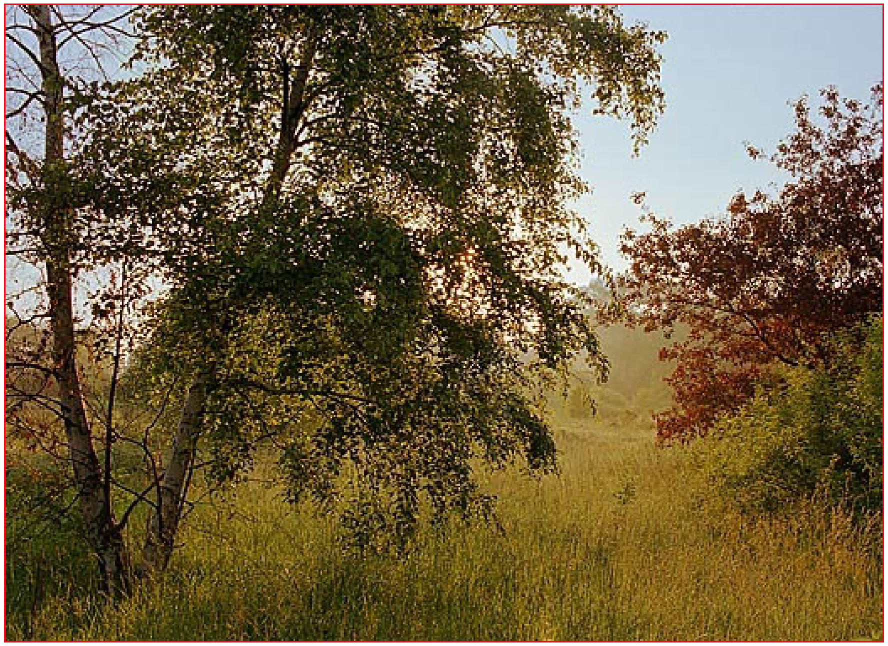
$$\frac{1}{9}\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}\right)\right]$$

$$= \frac{1}{9}\begin{bmatrix} -\alpha & -\alpha & -\alpha \\ -\alpha & 9+8\alpha & -\alpha \\ -\alpha & -\alpha & -\alpha \end{bmatrix}$$

Unsharp Masking Image



Original Image



After Unsharp Masking

Deconvolution

If we want to "undo" low-pass filter $H(u)$,

$$H_{inv}(u) = 1/H(u)$$

Problem 1: This assumes you know the point-spread function

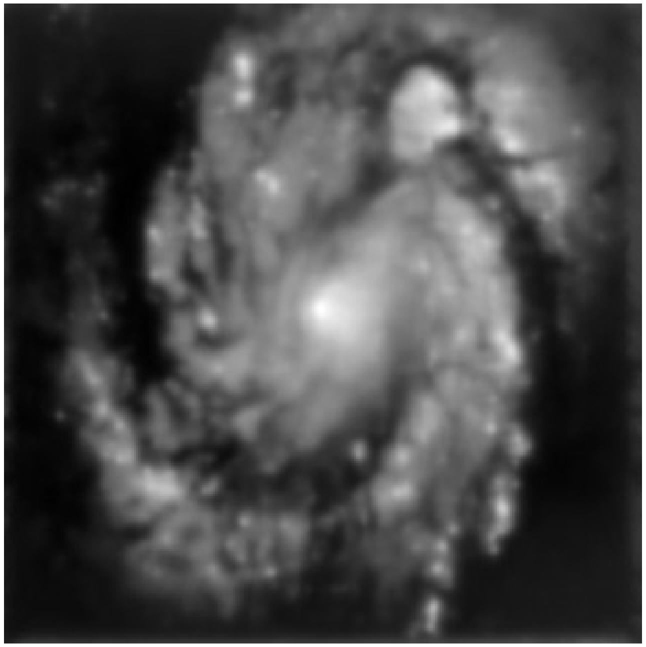
Problem 2: H may have had small values at high frequencies, so H_{inv} has large values (multipliers)

Small errors (noise, round-off, quantization, etc.) can get magnified greatly, especially at high frequencies

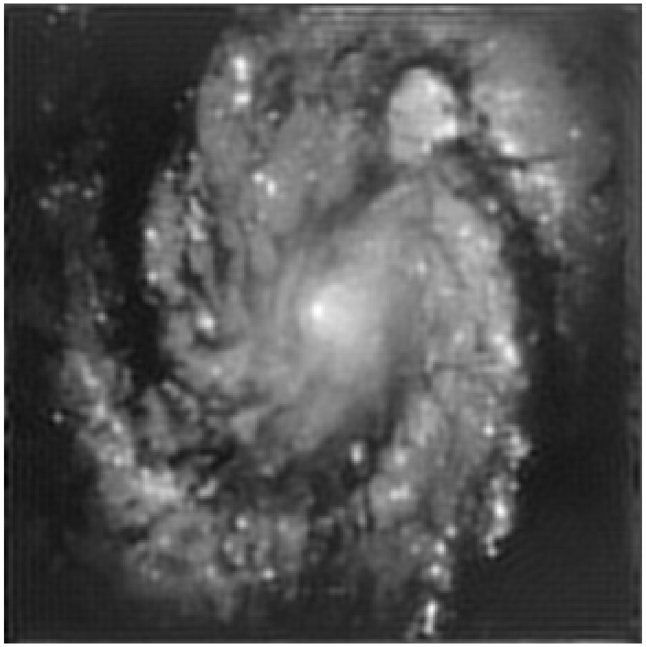
This is a common problem for all high-pass methods

Example: Deconvolution

Early Hubble space telescope image with precisely known optical aberrations



Before deconvolution



After deconvolution

Band-Pass Filtering

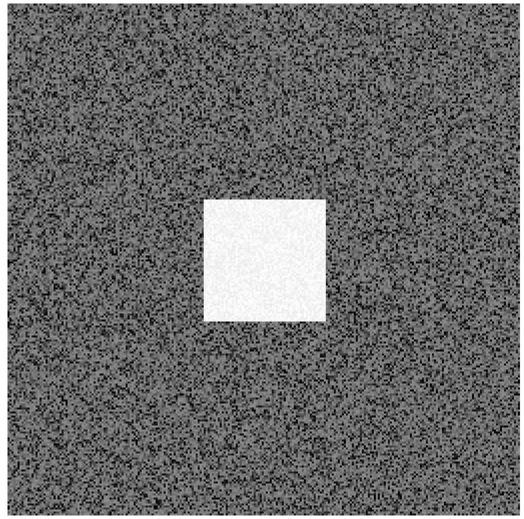
Tradeoff: Blurring vs. Noise

- Low-Pass: reduces noise but accentuates blurring
- High-Pass: reduces blurring but accentuates noise

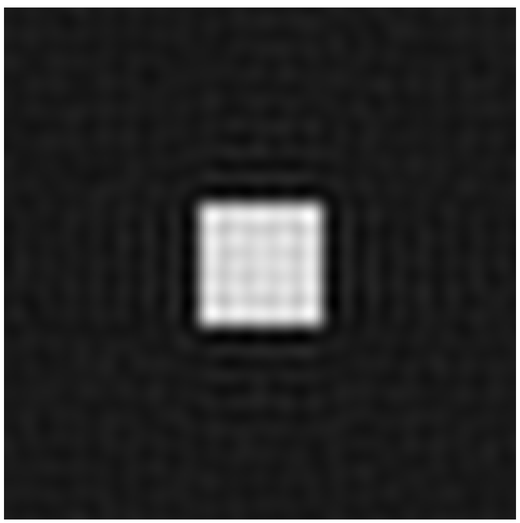
A compromise:

Band-pass filtering boosts certain midrange frequencies and partially corrects for blurring, but does not boost the very high (most noise corrupted) frequencies

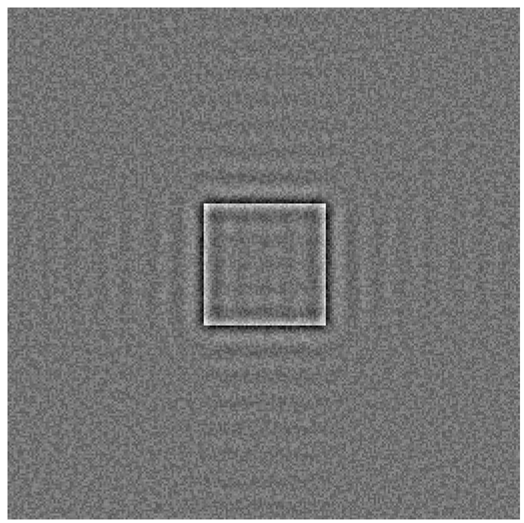
Band-Pass Filtering vs. Low-Pass, High-Pass Filtering



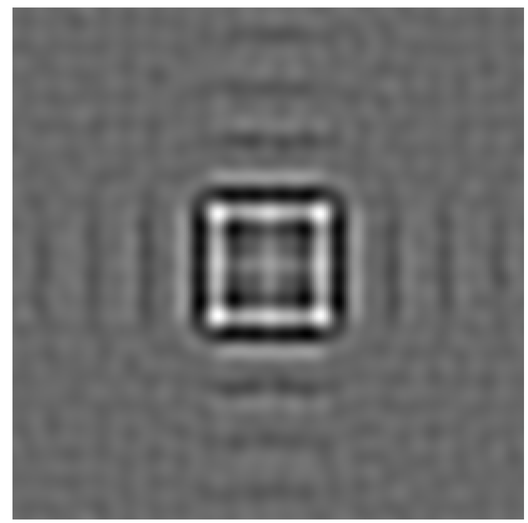
Original Image



After Low-pass filter



After High-pass filter



After Band-pass filter

Median "Filtering"

Instead of a local neighborhood weighted average, compute the *median* of the neighborhood

- Advantages:
  - Removes noise like low-pass filtering does
  - Value is from actual image values
  - Removes outliers – doesn't average (blur) them into result ("despeckling")
  - Edge preserving
- Disadvantages:
  - Not linear
  - Not shift invariant
  - Slower to compute

Median "Filtering"

Original image

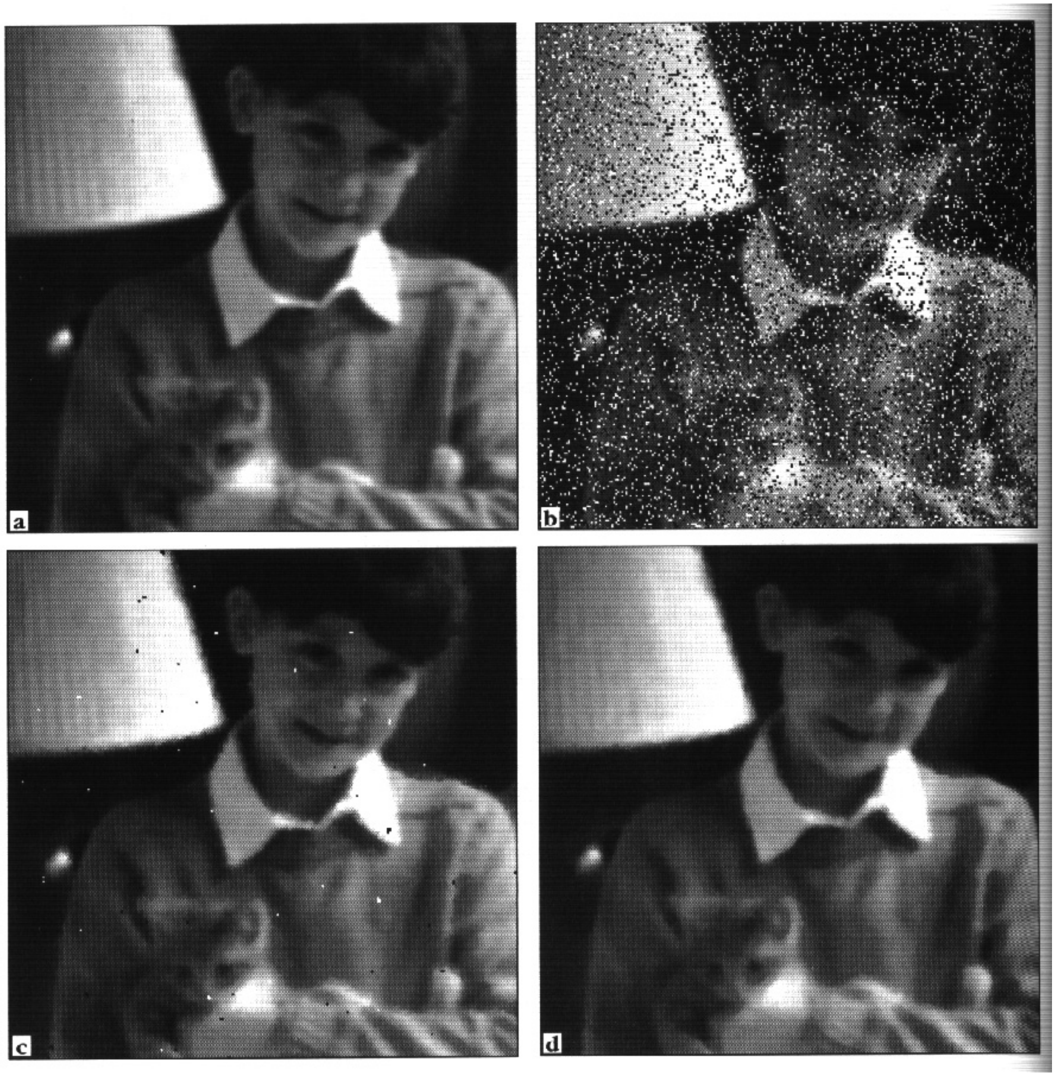


Image a with 10% of the pixels randomly selected and set to black, and another 10% randomly selected and set to white

Application of median filtering to image b using a 3x3 square region

Application of median filtering to image b using a 5x5 square region

Removal of shot noise with a median filter

Figure and Text Credits

Text and figures for this lecture were adapted in part from the following source, in agreement with the listed copyright statements:

http://web.engr.oregonstate.edu/~enm/cs519

Resources

Textbook:

Kenneth R. Castleman, Digital Image Processing, Chapter 11