Transform Methods & Signal Processing

lecture 06

Matthew Roughan
<matthew.roughan@adelaide.edu.au>

Discipline of Applied Mathematics
School of Mathematical Sciences
University of Adelaide

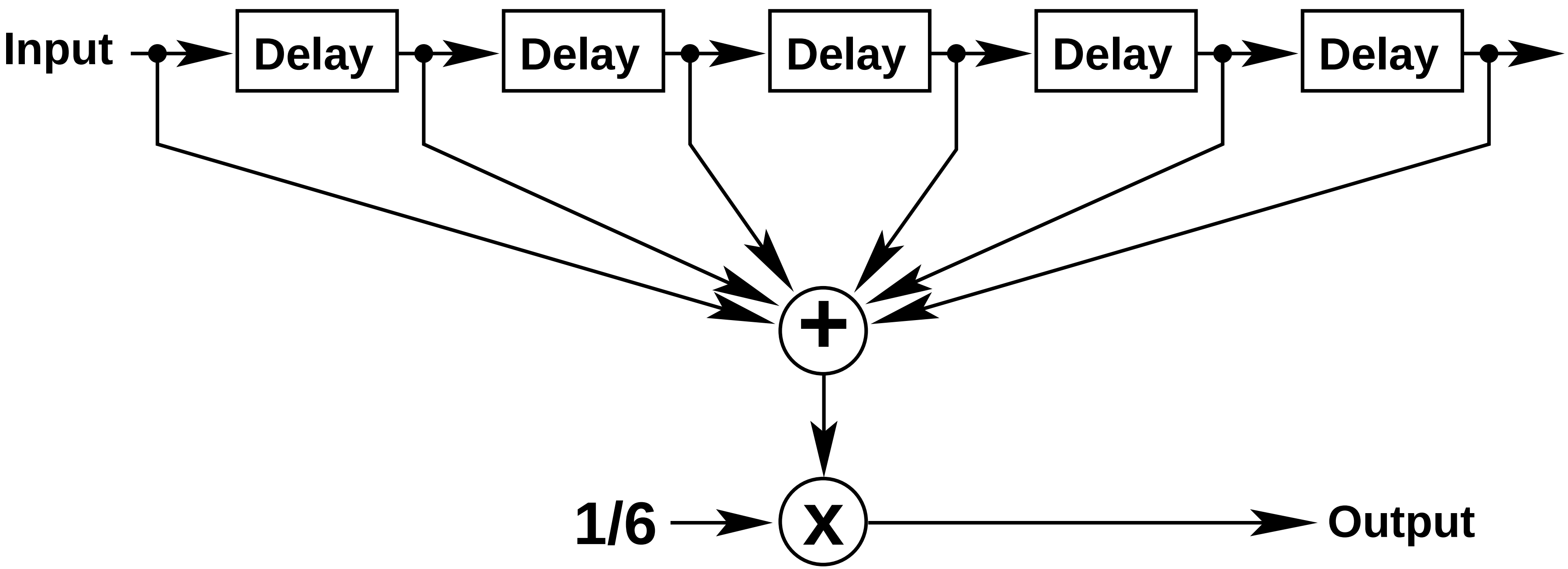
July 27, 2009

Block Diagrams

Block diagrams provide a visual metaphor for filters that can be useful in the design and analysis of filters. We can put together a more complex filter from simpler components as it we were putting together Lego.

How to draw filters

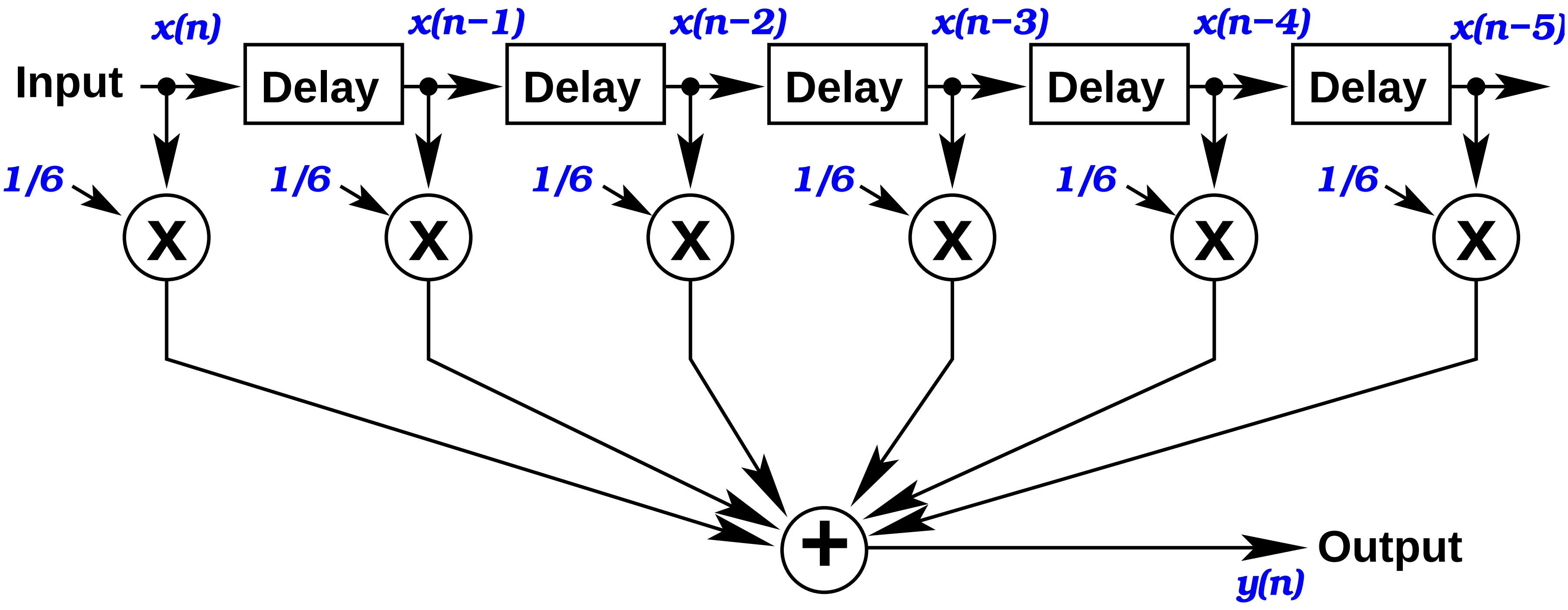
Block diagrams: example $y(n) = \frac{1}{6}\sum_{i=0}^{5} x(n-i)$



A six tap filter.

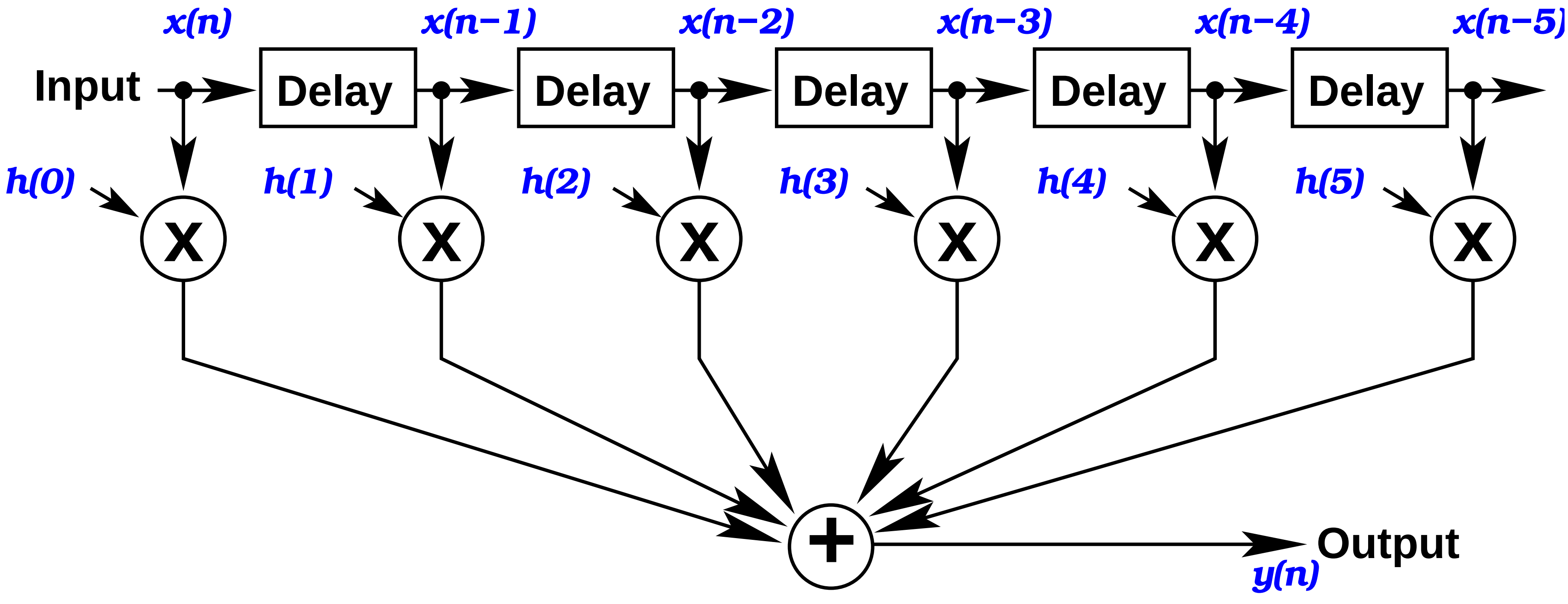
How to draw filters

Block diagrams: example $y(n) = \sum_{i=0}^{5} \frac{1}{6} x(n-i)$



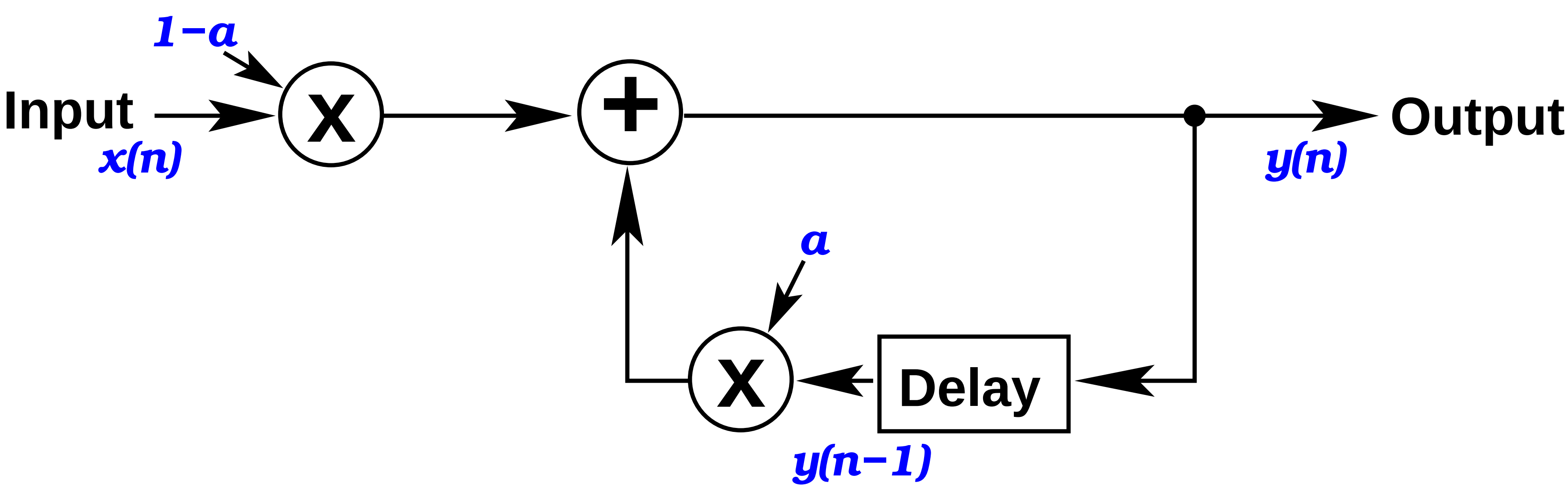
Block diagrams of FIR filters

FIR filter: $y(n) = \sum_{i=0}^{5} h(i)x(n-i)$



Block diagrams of IIR filters

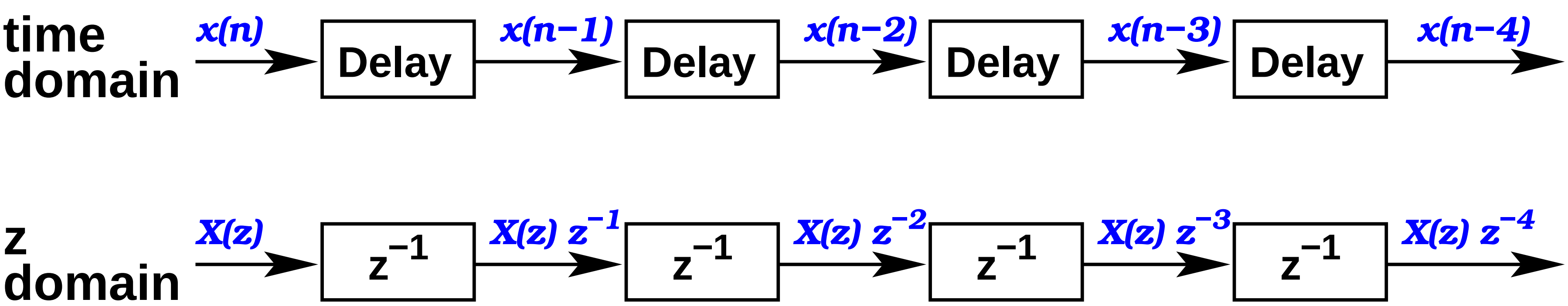
IIR filter: $y(n) = ay(n-1) + (1-a)x(n)$



IIR filter designs use feedback (or recursion)

Block diagrams and Z-transforms

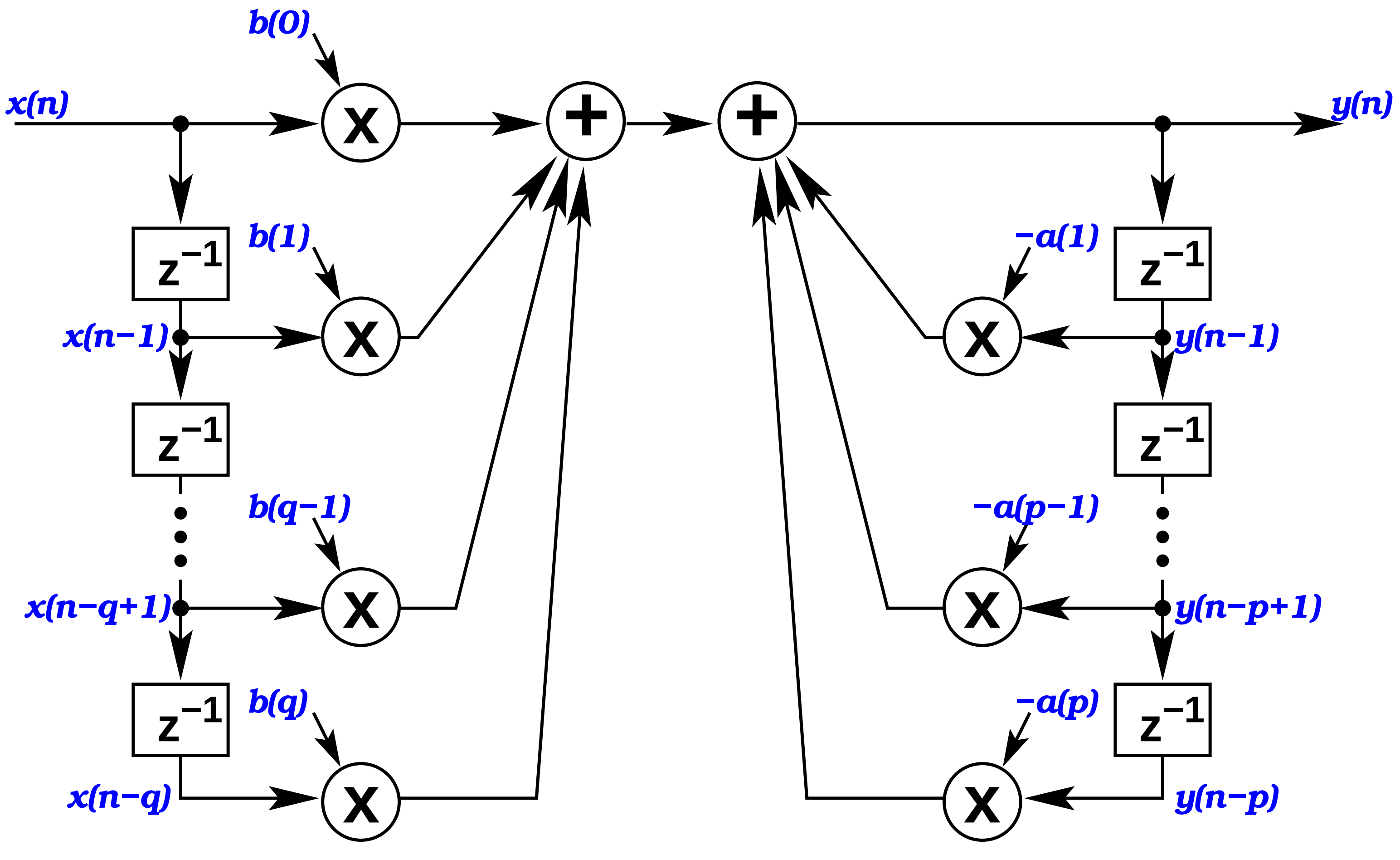
Notice that if $y(n) = x(n-1)$ then $Y(z) = z^{-1}X(z)$



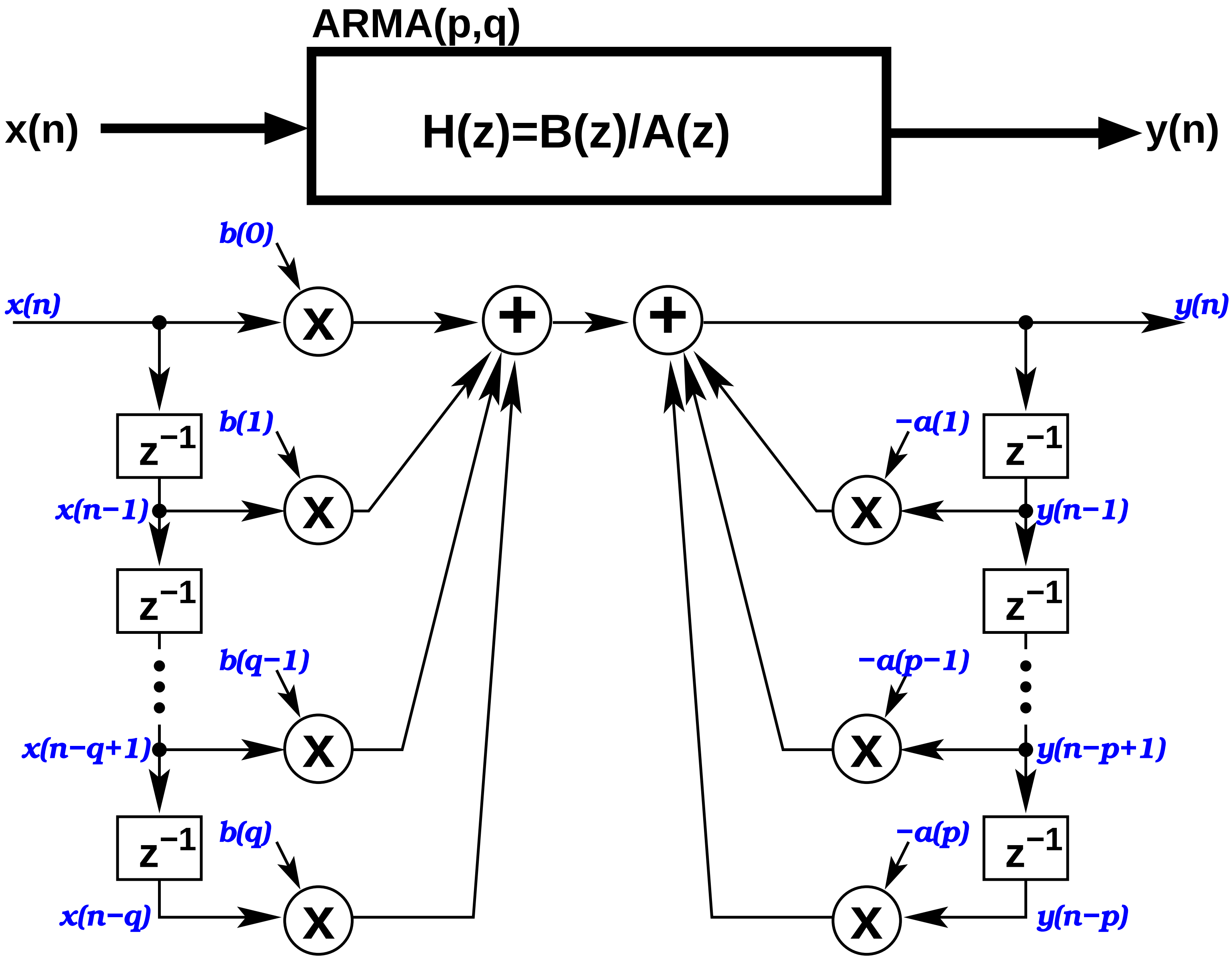
So we can represent block diagrams in this fashion.

Block diagram of ARMA

ARMA: $y(n) = -\sum_{i=1}^{p} a(i)y(n-i) + \sum_{i=0}^{q} b(i)x(n-i)$

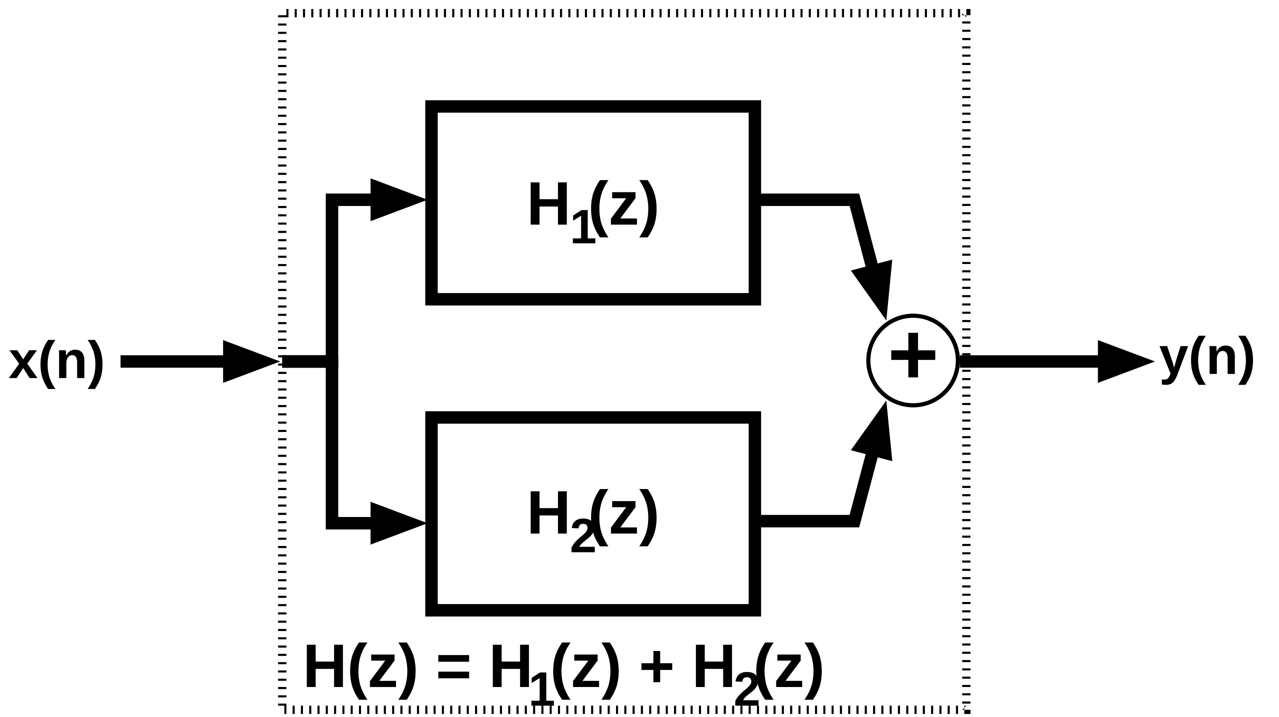


Abstracting the blocks



Connecting the blocks

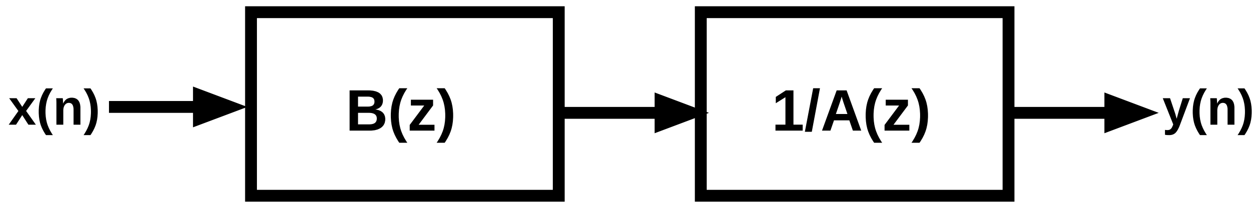
In parallel



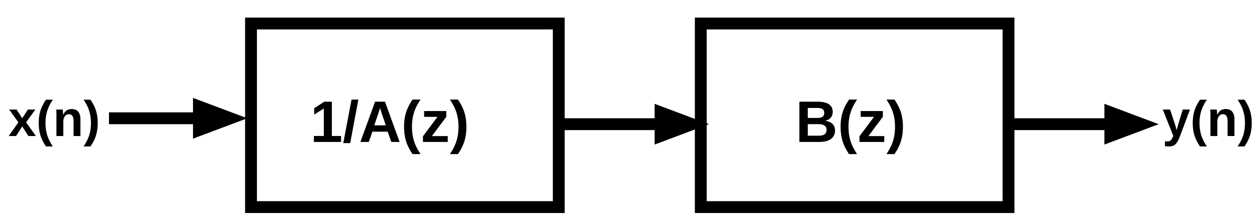
$$Y(z) \quad = \quad [H_1(z) + H_2(z)]\, X(z)$$

Connecting the blocks

Series, or cascade

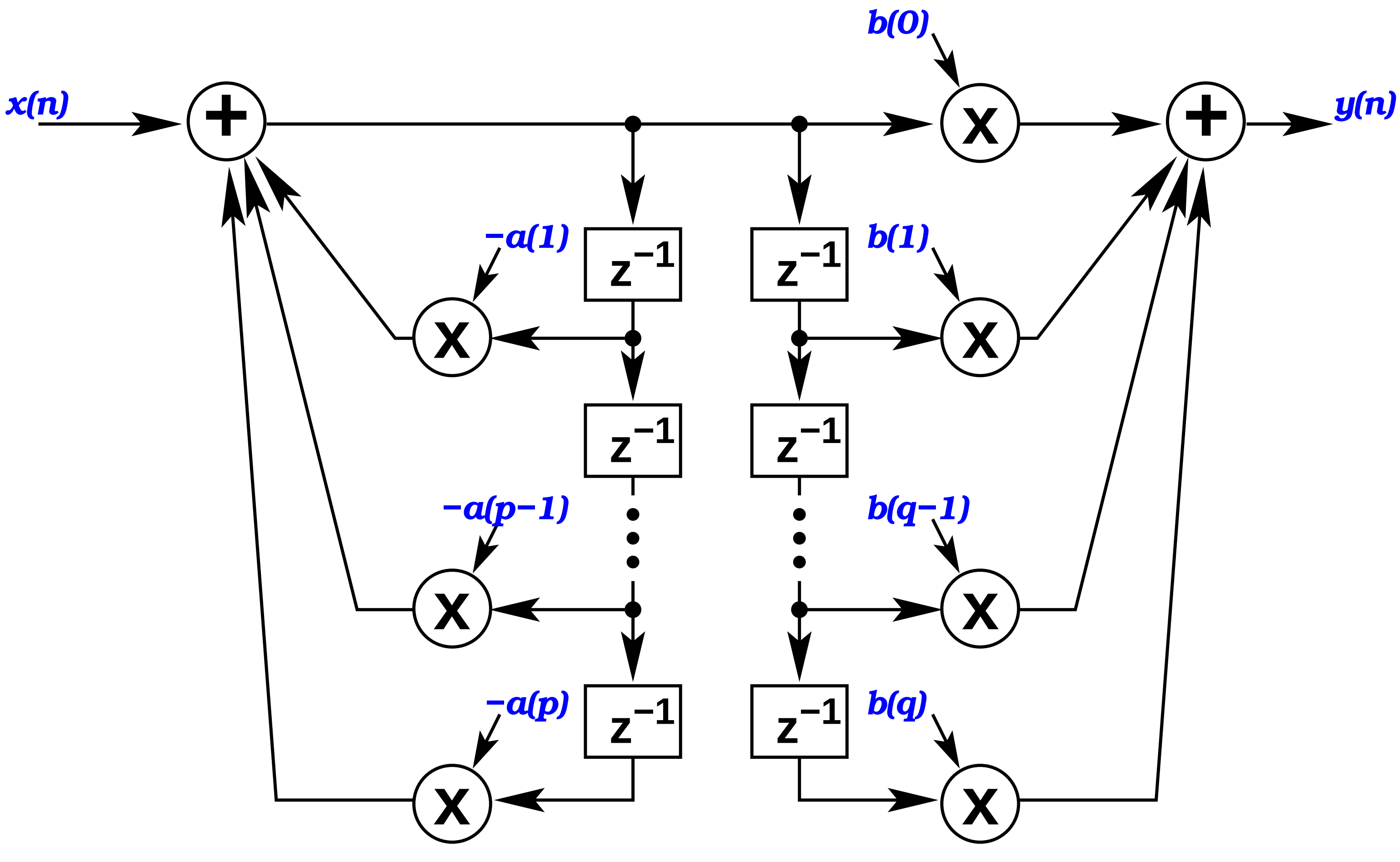


$$
\begin{aligned}
Y(z) &= \frac{1}{A(z)} \times B(z) \times X(z) \\
&= \frac{B(z)}{A(z)} X(z) \\
Y(z) &= B(z) \times \frac{1}{A(z)} \times X(z)
\end{aligned}
$$



Transposed Block diagram of ARMA

ARMA: $$y(n) = -\sum_{i=1}^{p} a(i)y(n-i) + \sum_{i=0}^{q} b(i)x(n-i)$$



Connecting the blocks

How do we use cascades?

- Can break a high order IIR filter into a series of cascades (of small order)
- Equivalent to factorizing the polynomial $A(z)$, e.g.

$$H(z) = B(z)\frac{1}{A(z)} = B(z)\prod_{i=1}^{n}\frac{1}{A_i(z)}$$

- can do similarly to FIR part $B(z)$
- note that degree of subparts sums to degree of combination

Filtering in the real world

Some things to be careful about

- **Coefficient quantization:** coefficients $a(i)$, $b(i)$ are also quantized. This shifts filter poles and zeros, possibly causing instability, or other problems.
- **Overflow:** intermediate values may overflow dynamic range, resulting in clipping, even if input and output lie within dynamic range
- use cascades to make design easier (and control of above effects)

Upsampling again

DFT properties: similarity

We can interleave a sequence with zeros, e.g.

$$y(n) = \begin{cases} x(n/K), & \text{if } n = 0, K, 2K, \ldots, (N-1)K \\ 0, & \text{otherwise} \end{cases}$$

The resulting DFT is

$$\mathcal{F}\{y\} = Y(k) = \begin{cases} X(k) & k = 0, \ldots, N-1 \\ X(k-N) & k = N, \ldots, 2N-1 \\ \vdots & \\ X(k-(K-1)N) & k = (K-1)N, \ldots, KN-1 \end{cases}$$

Upsampling again

Practical use: upsampling (interpolation)

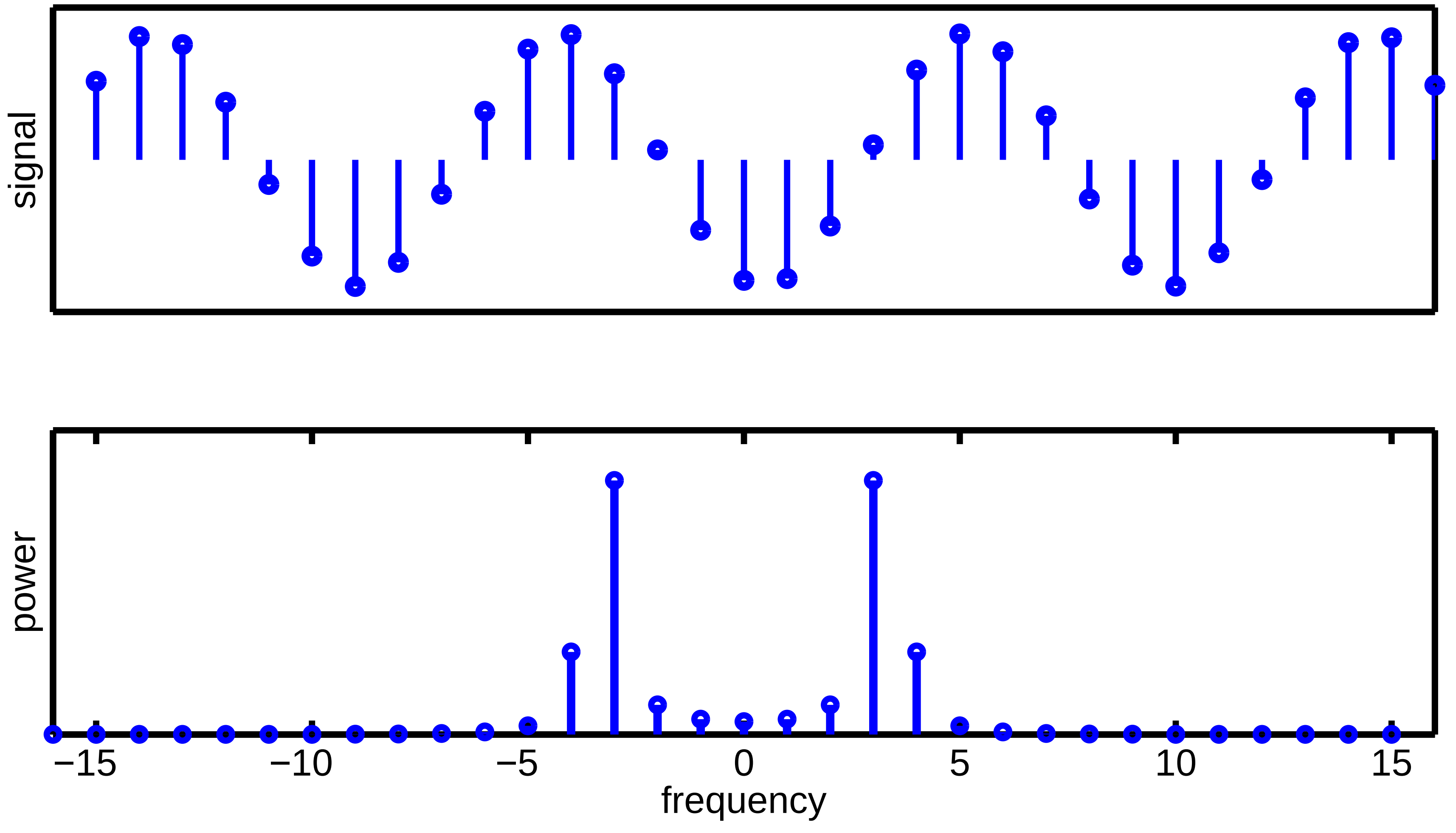
> We have a sequence sampled every t_s seconds, e.g. at a rate $f_s = 1/t_s$, but we need a sequence sampled at rate $K f_s$.

Approach: produce a new sequence with $K-1$ zeros interleaved between each original data point.

- the frequency resolution doesn't change, but now we have K repeats of the original spectrum.
- to get a signal with the same original band-limited power-spectrum, we apply a low-pass filter, smoothing the data.

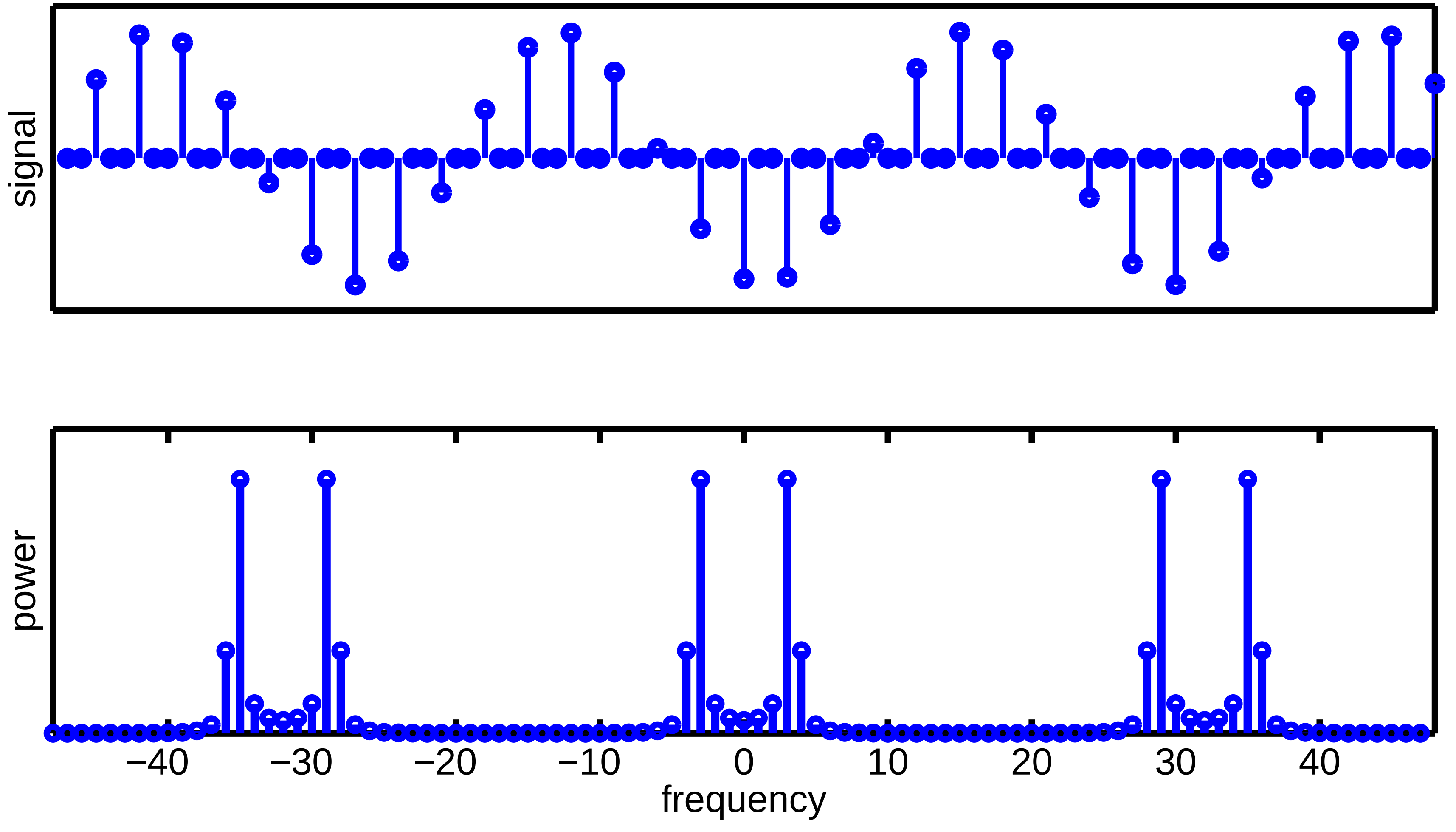
Upsampling example

32 samples (frequency 3.4 cycles)



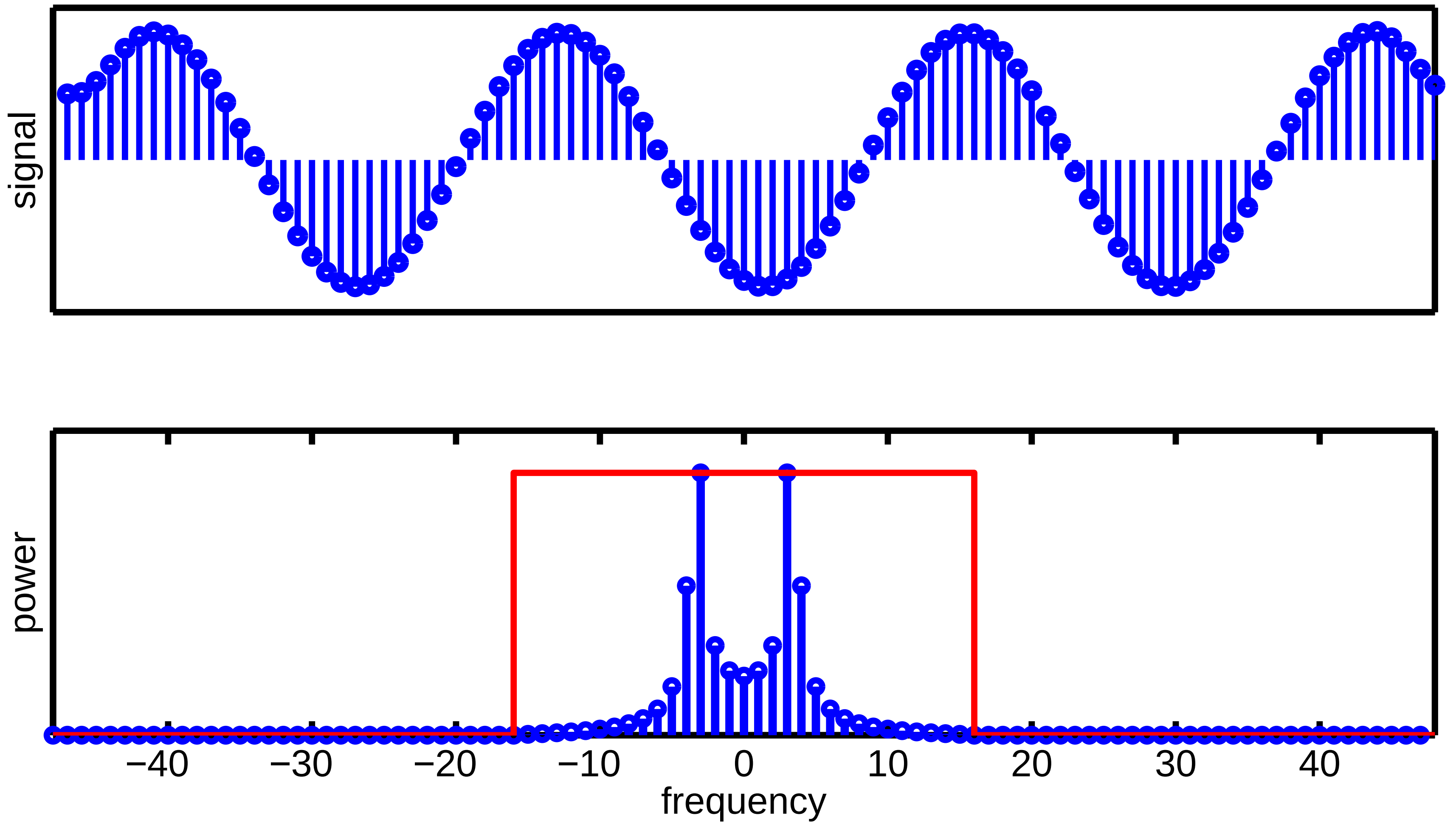
Upsampling example

3 $\times$'s upsampled (96 samples)

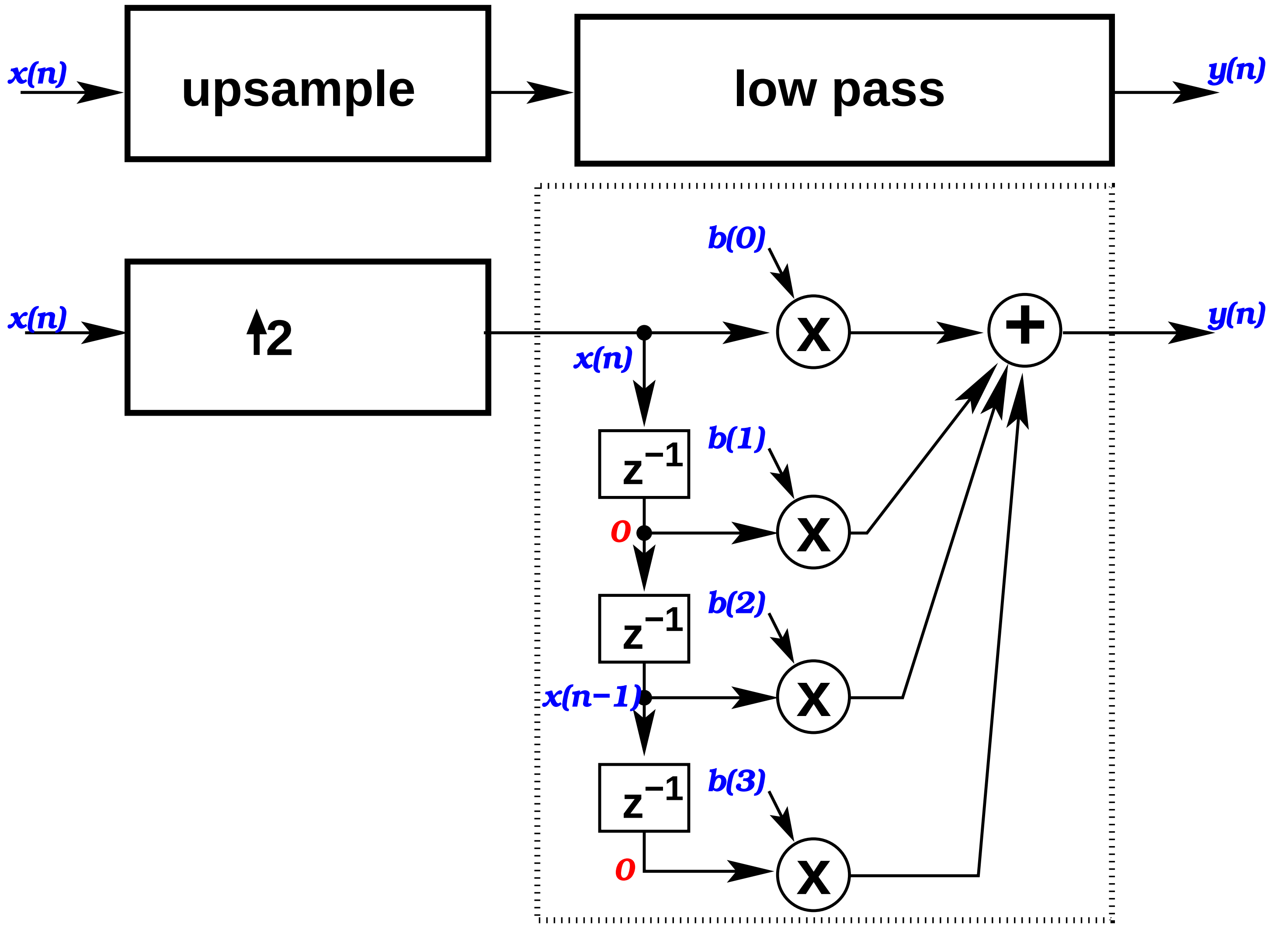


Upsampling example

low pass filter, then IDFT

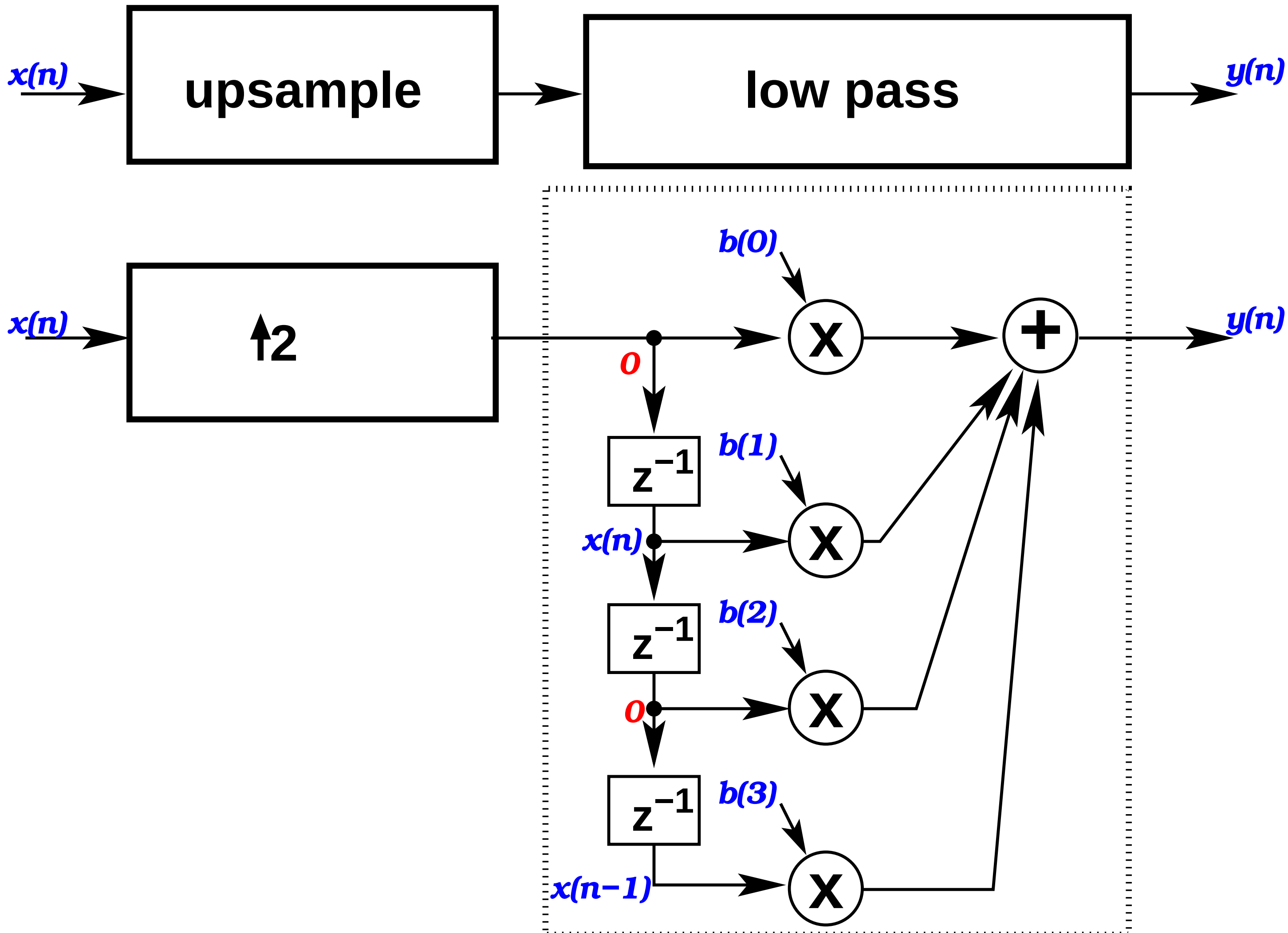


Upsampling block diagram



Notice lots of multiplies by zero

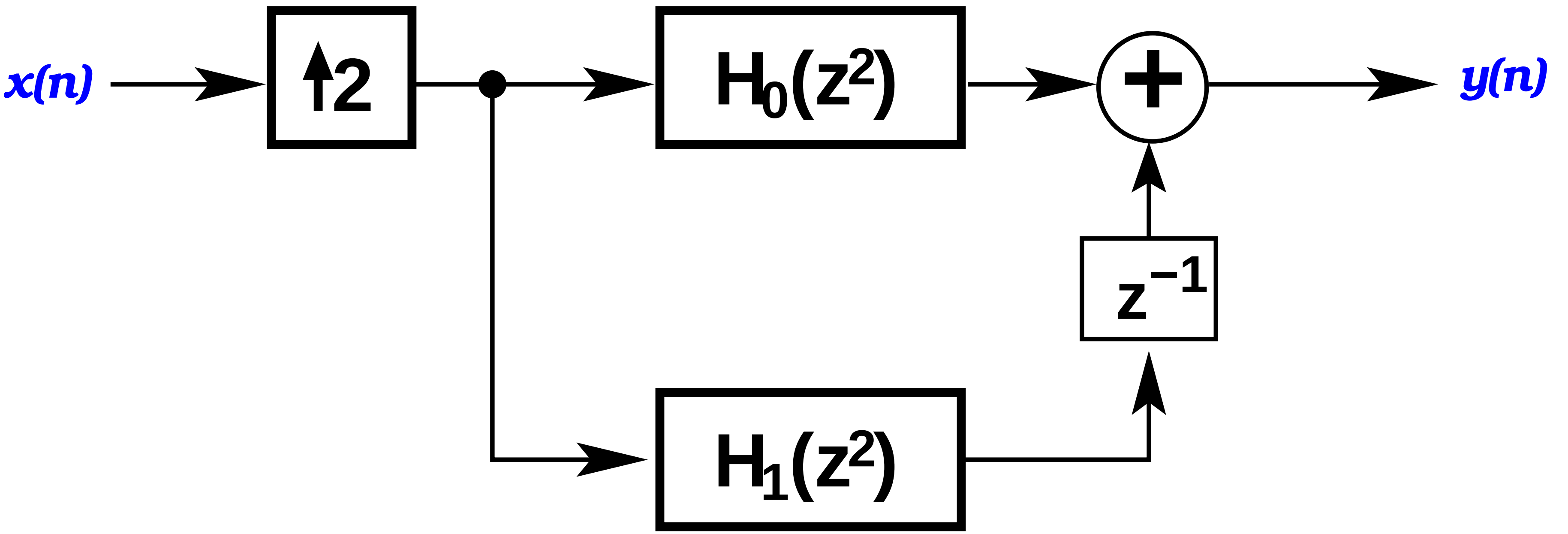
Upsampling block diagram



Notice lots of multiplies by zero

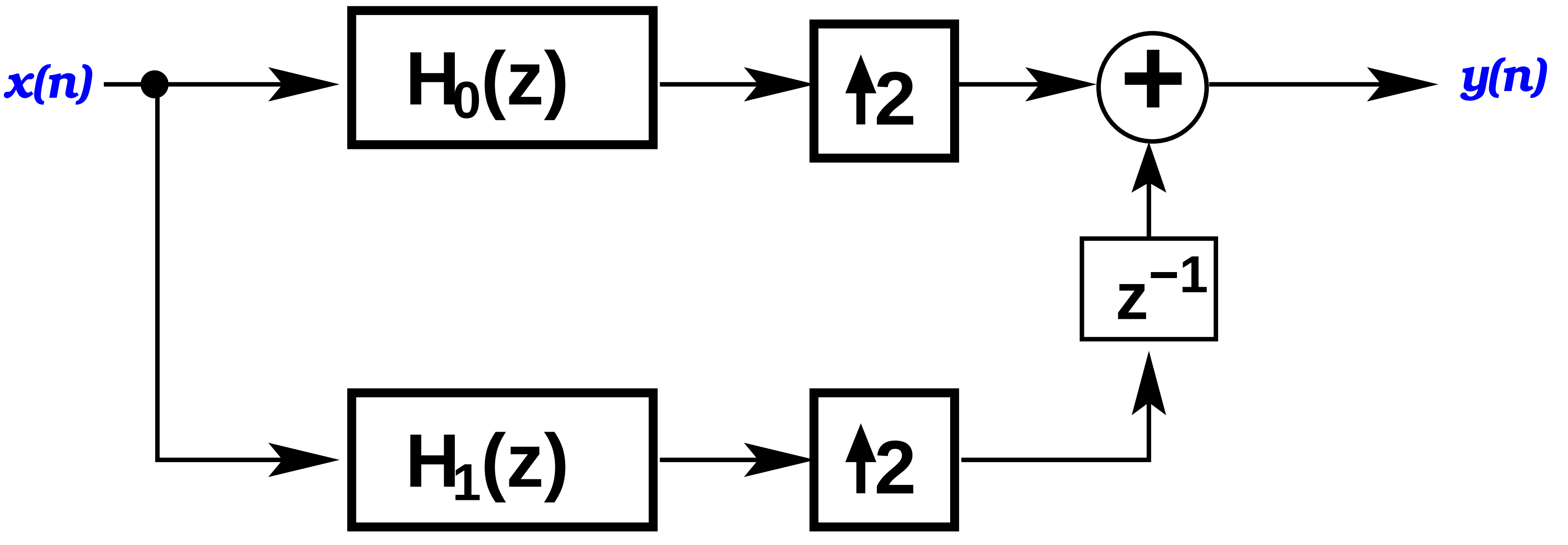
Upsampling block diagram

Trick 1: separate into two separate filters.



Upsampling block diagram

Trick 2: low-pass before upsampling.



Downsampling

We can downsample by simply dropping data points, but this could cause aliasing (as the new critical frequency will depend on the downsampled rate).

As with Analogue to Digital conversion we must low-pass the signal before downsampling.

We can pull the same trick (as with upsampling) to filter at the lower rate after downsampling.

Resampling

Can do resampling by rational numbers q/p through upsampling by p, and then downsampling by q, but this is inefficient, and there are better approaches.

2D Filters

We can extend our work on filters into 2 dimensions

Convolution in 2D

Convolution generalizes to 2D, e.g., the two-dimensional convolution of continuous functions $f(x,y) = \delta(x)r(y)$ with $g(x,y) = r(x)\delta(y)$ is

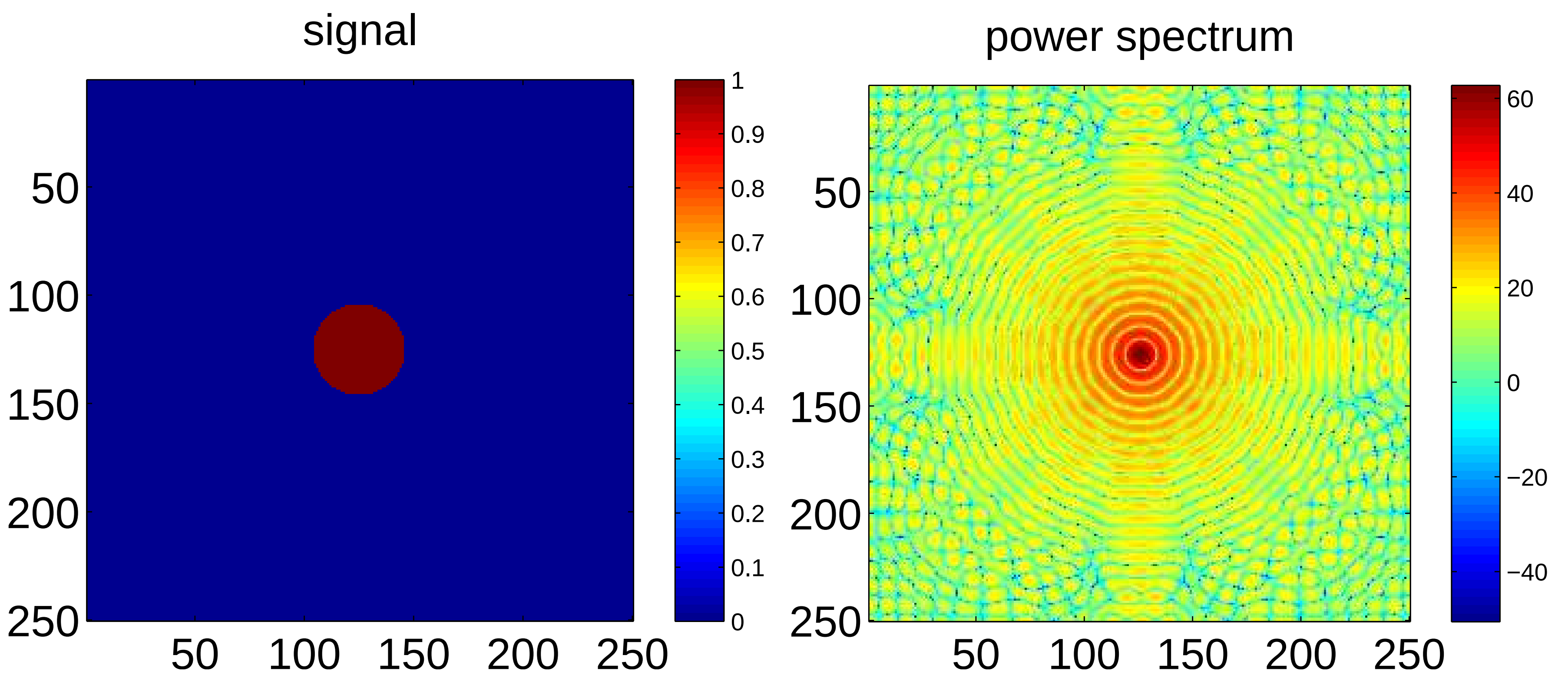
$$[f * g](x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x',y')g(x-x',y-y')\,dx'\,dy'$$

Likewise, for discrete signals we can extend the idea of a LTI filter (a convolution) to 2D

$$[x * y](n,m) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k,l)y(n-k,m-l)$$

Filters in 2D

2D data



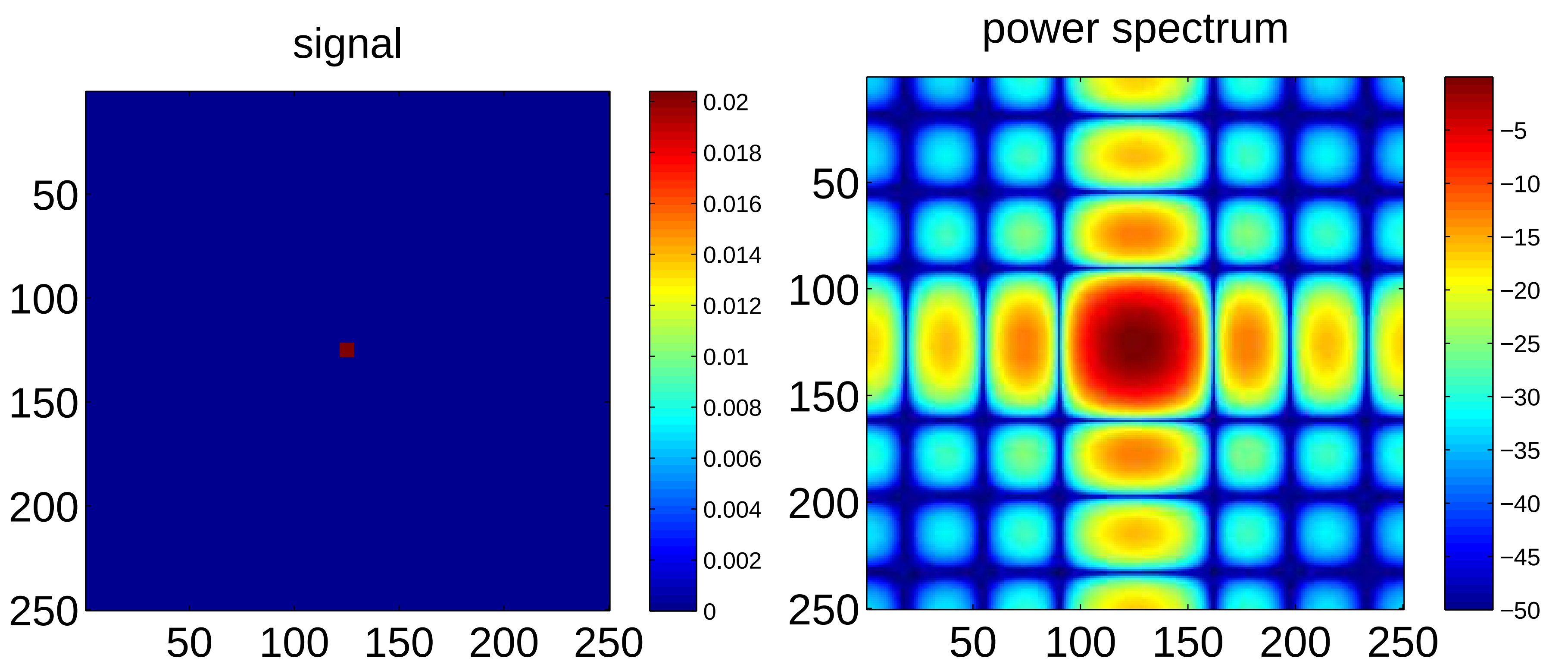
Filters in 2D

Spatial rectangular low-pass

$$\frac{1}{49}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

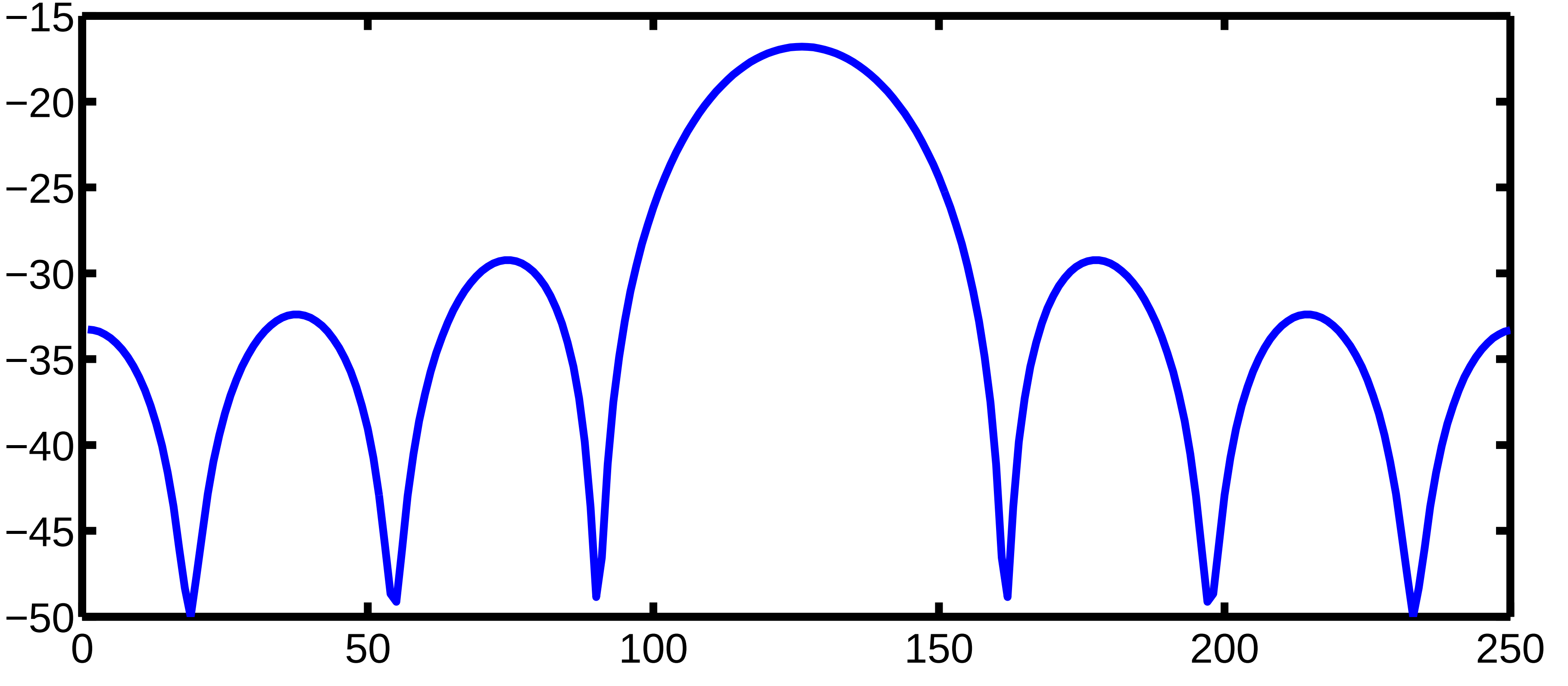
Filters in 2D

Spatial rectangular low pass, frequency response



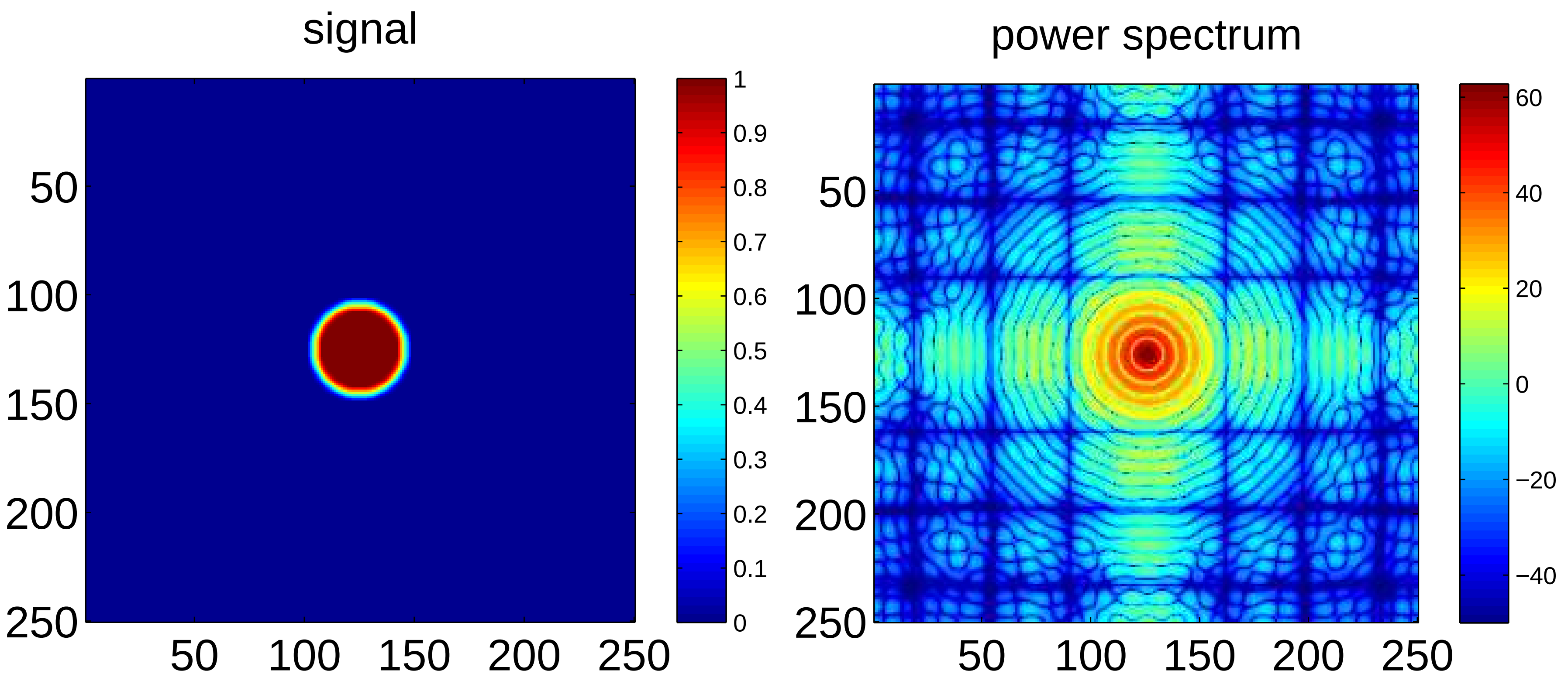
Filters in 2D

Spatial rectangular low pass, frequency response 1D



Filters in 2D

Spatial rectangular low pass



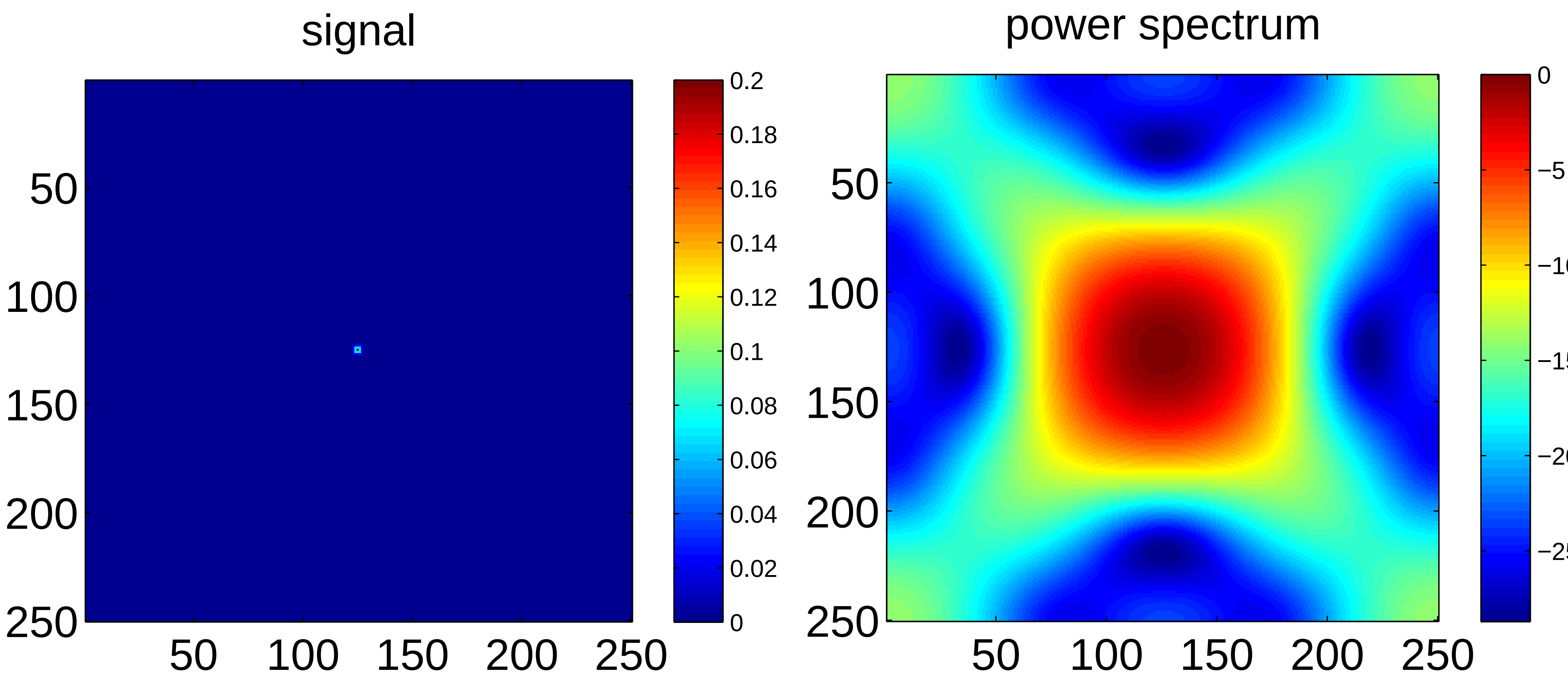
Filters in 2D

Approximately Gaussian low-pass

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 4 & 12 & 4 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

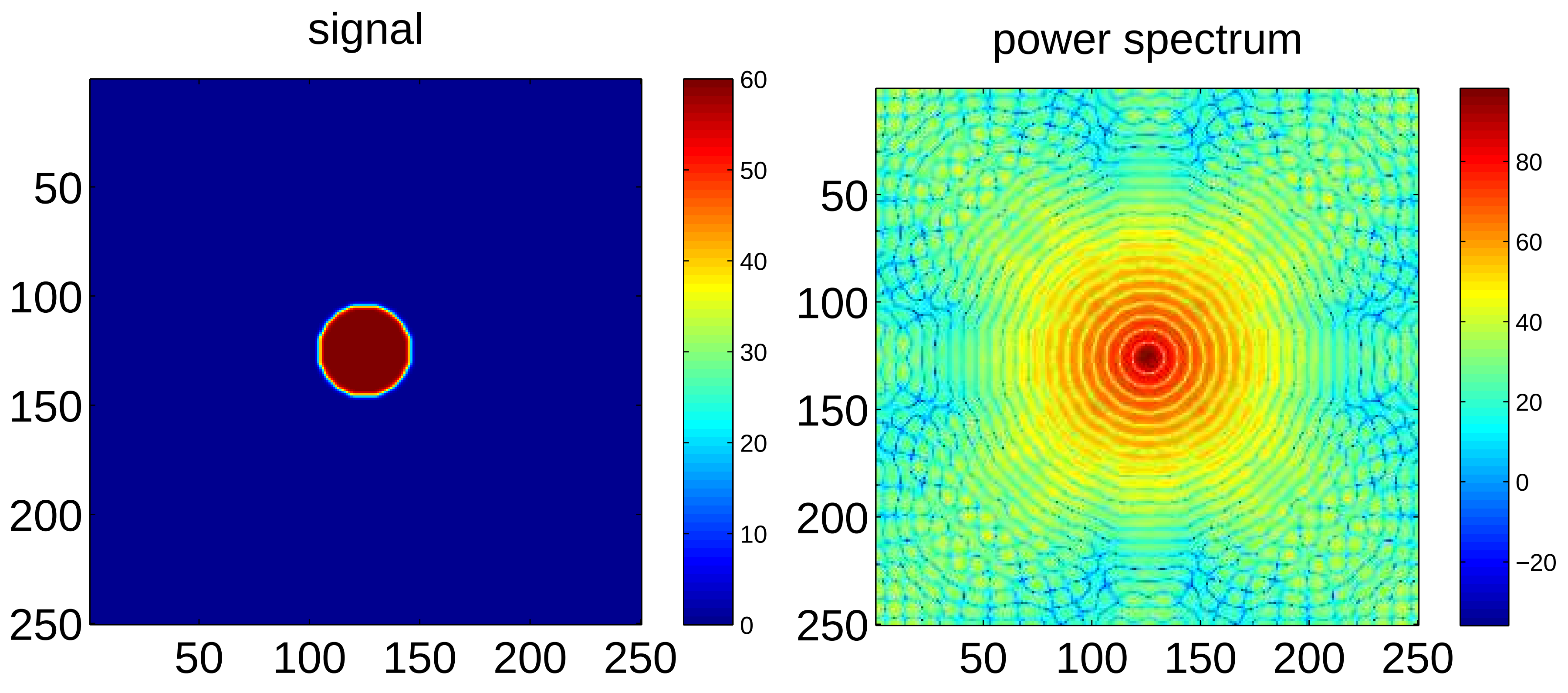
Filters in 2D

Approximately Gaussian low-pass, frequency response



Filters in 2D

Approximately Gaussian low-pass



Edge detection

Edge detection is a high pass operation, but there are a number of ways to do this in 2D.

- gradient: look for max and min in gradients in image
  - Sobel
  - Prewitt
  - Roberts
- Laplacian: look for zeros of second derivative
  - Marrs-Hildreth

Sobel Edge detection

Look for vertical and horizontal edge separately, using filters

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

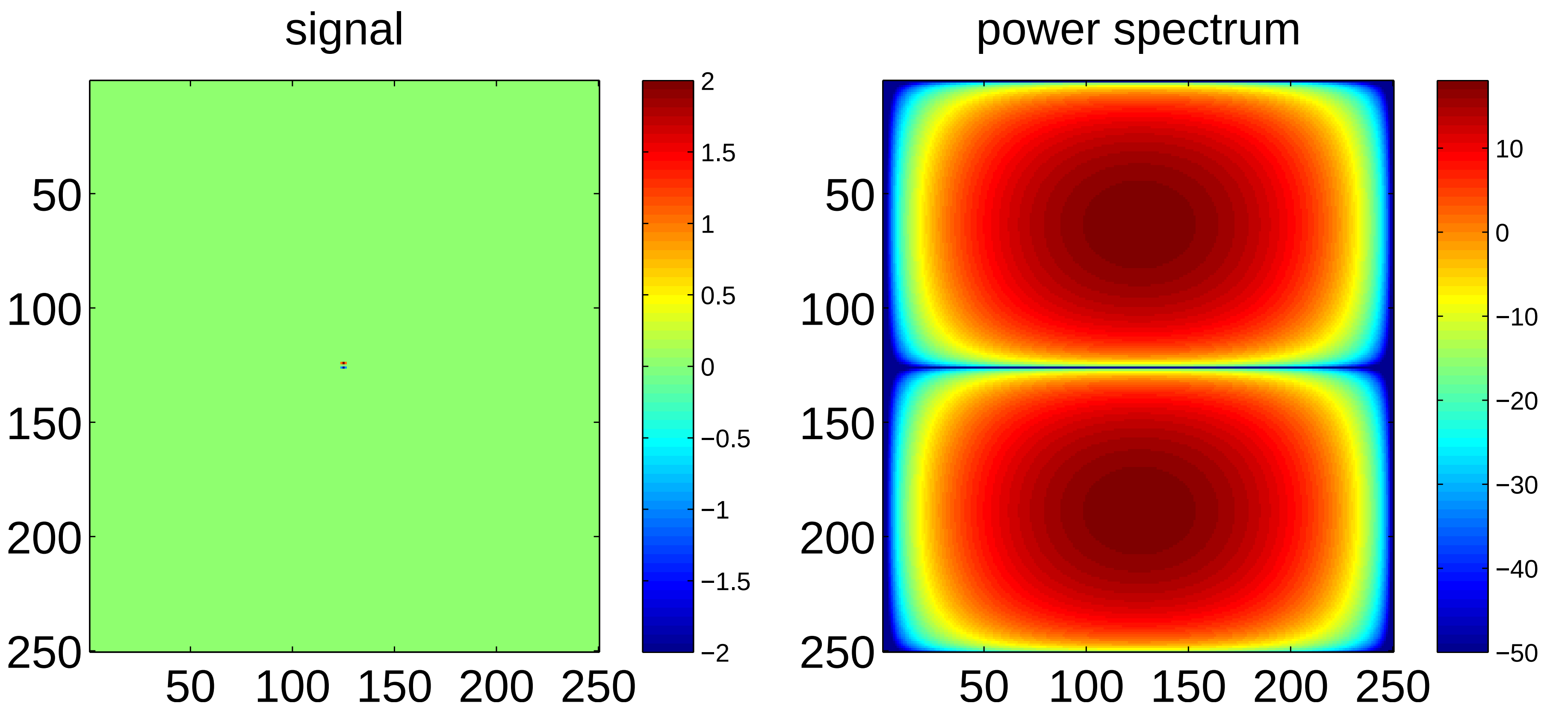
The threshold for values above or below T.

Alternatively, can combine to get edge magnitude and direction by

$$\begin{aligned} M_{\text{sobel}} &= \sqrt{M_{\text{vertical}}^2 + M_{\text{horizontal}}^2} \\ \phi_{\text{sobel}} &= \tan^{-1}\left(M_{\text{vertical}}/M_{\text{horizontal}}\right) \end{aligned}$$

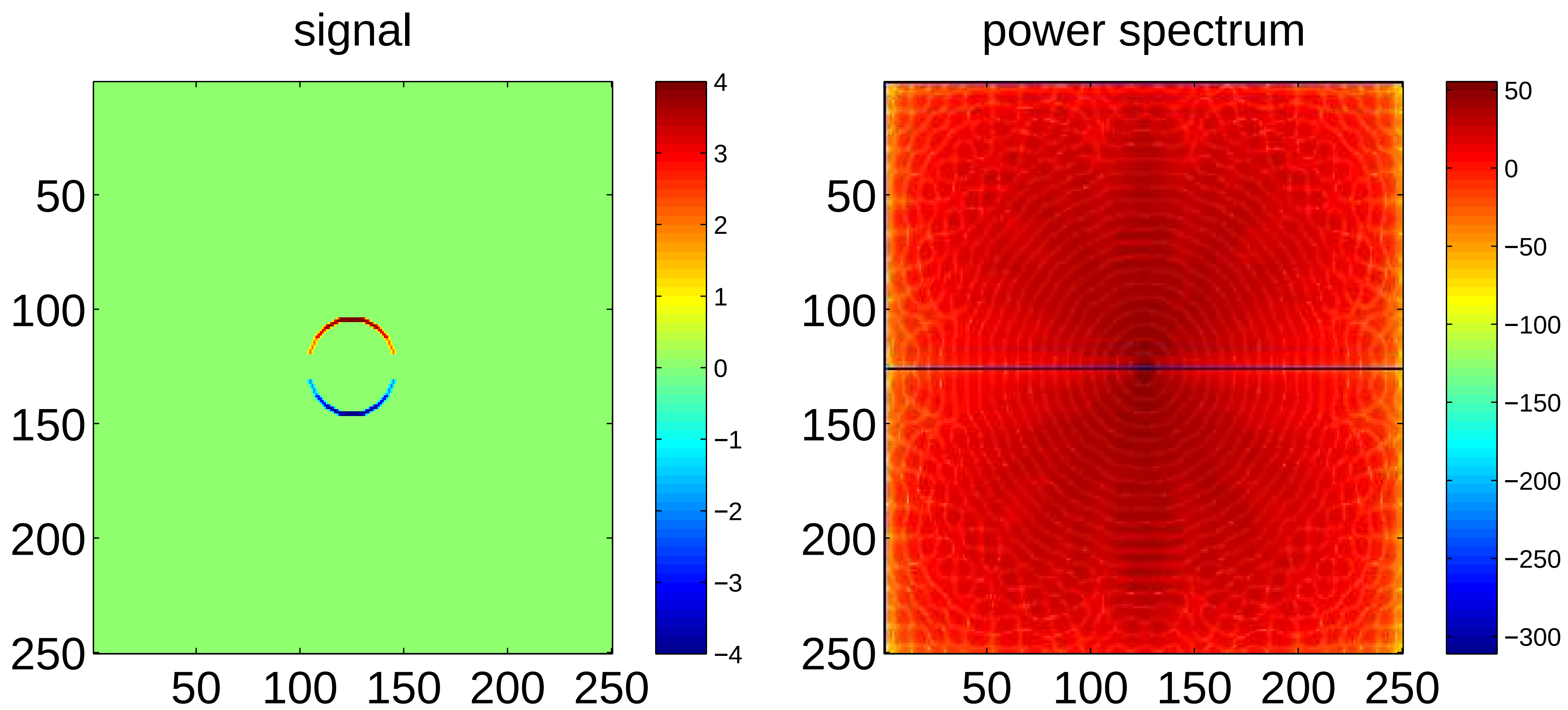
Filters in 2D

Sobel edge detection, freq. response



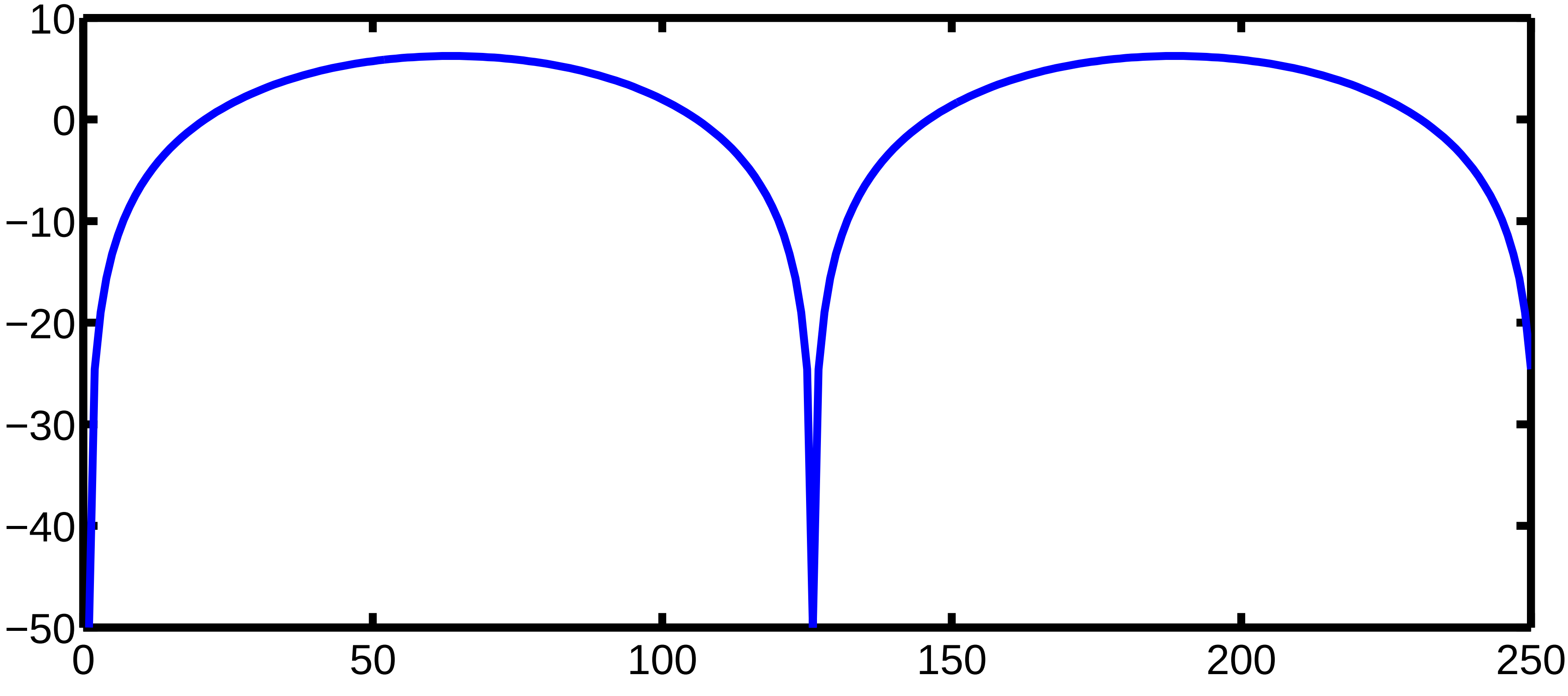
Filters in 2D

Sobel edge detection applied to image.



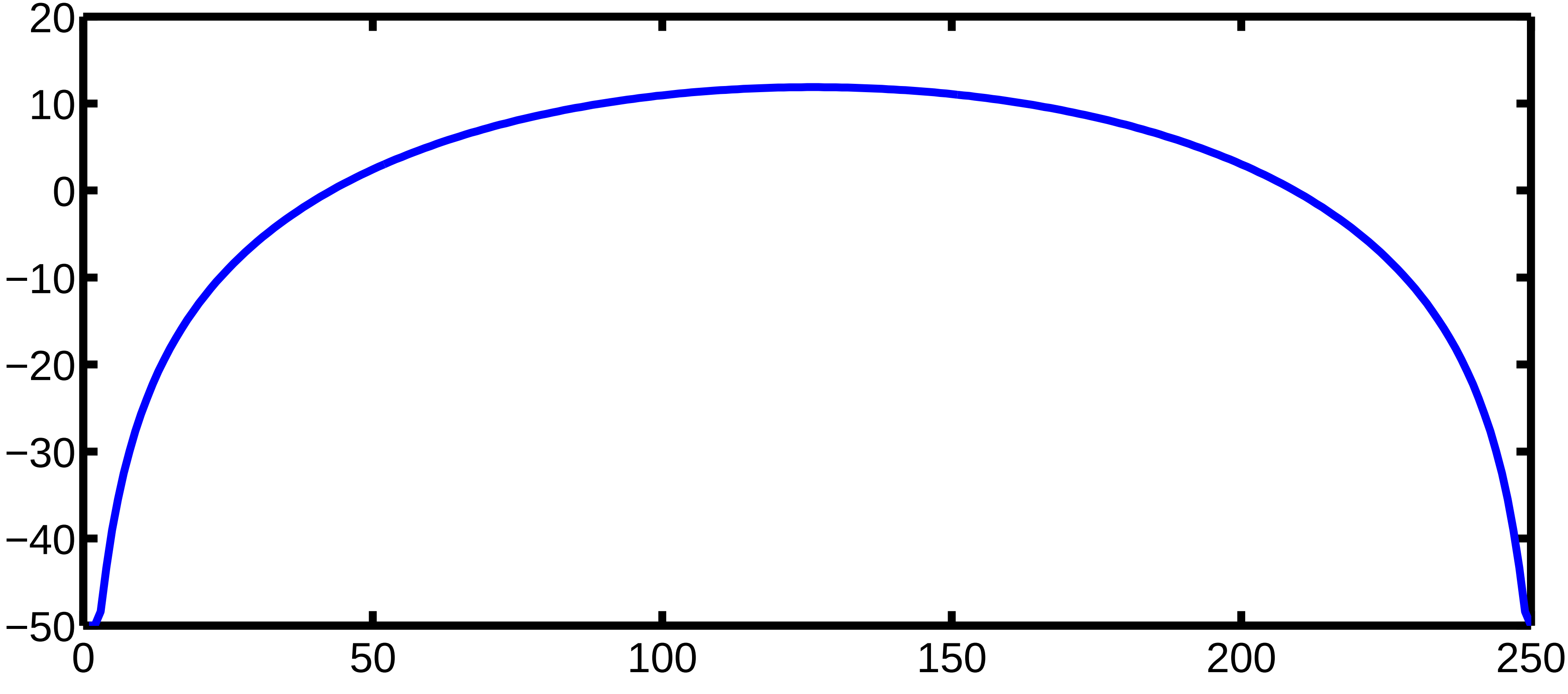
Filters in 2D

Vertical high pass (Sobel edge detection), freq. response, horizontal



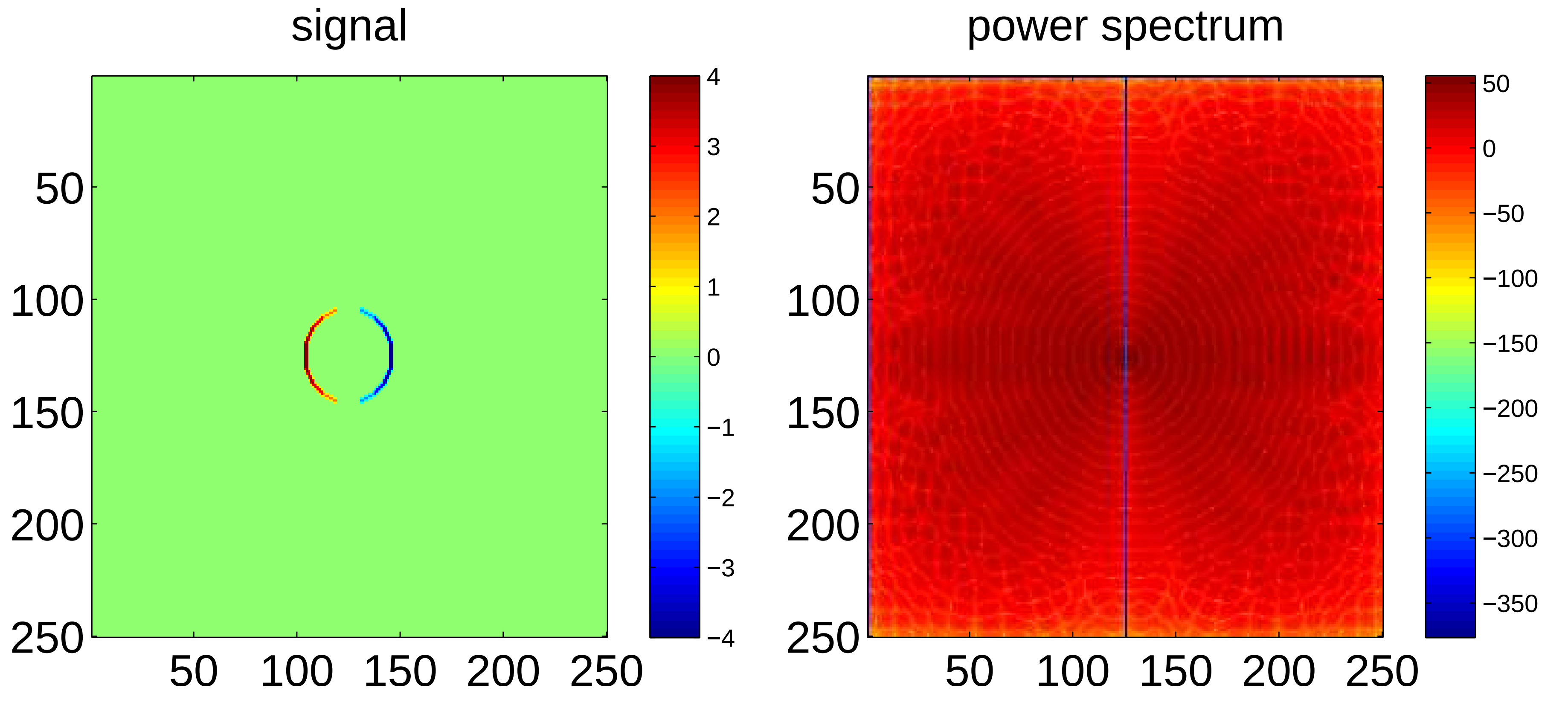
Filters in 2D

Vertical high pass (Sobel edge detection), freq. response, vertical



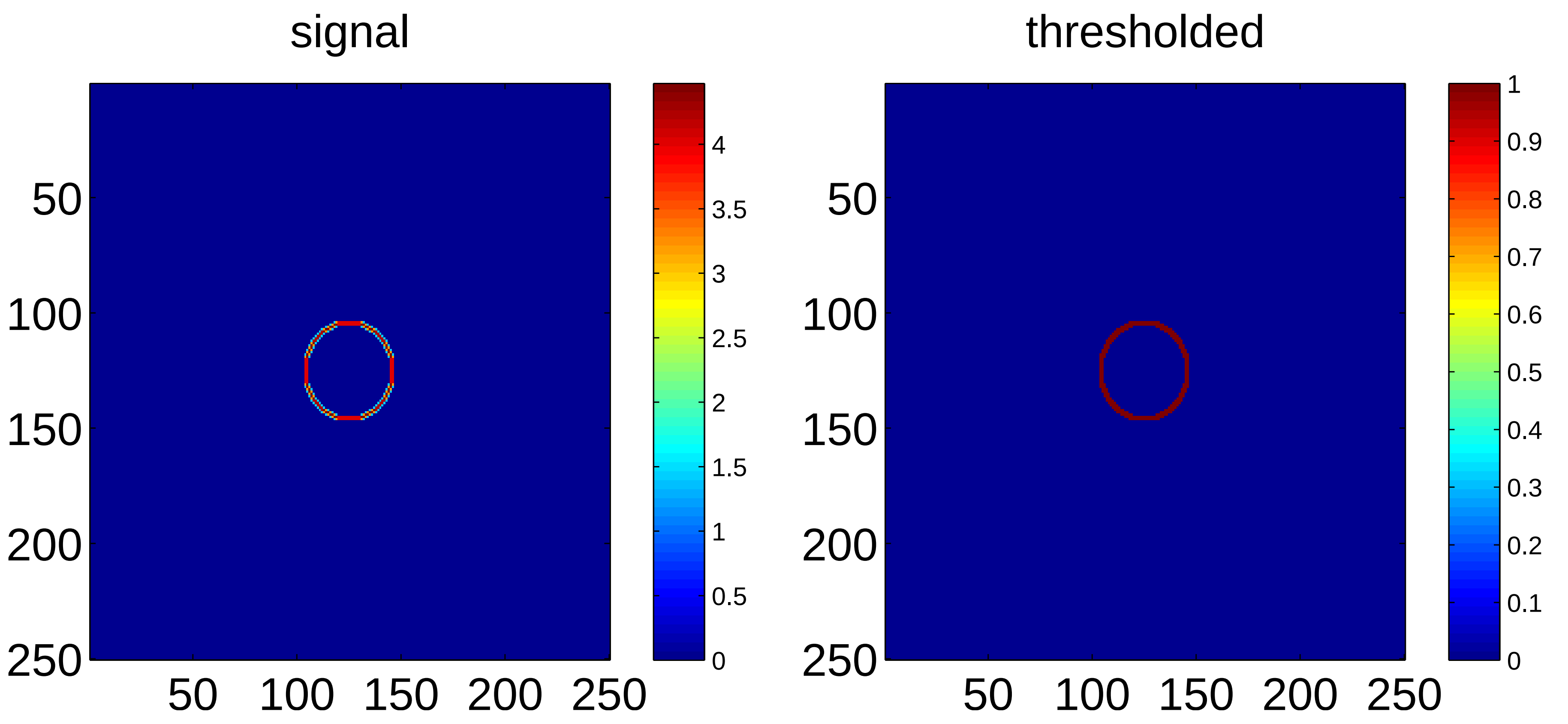
Filters in 2D

Horizontal high pass (Sobel edge detection)



Filters in 2D

Combined Sobel filters, and thresholded version



Prewitt filters

Look for vertical and horizontal edge separately, using filters

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Roberts filters

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Laplacian

Approximation of $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

3×3

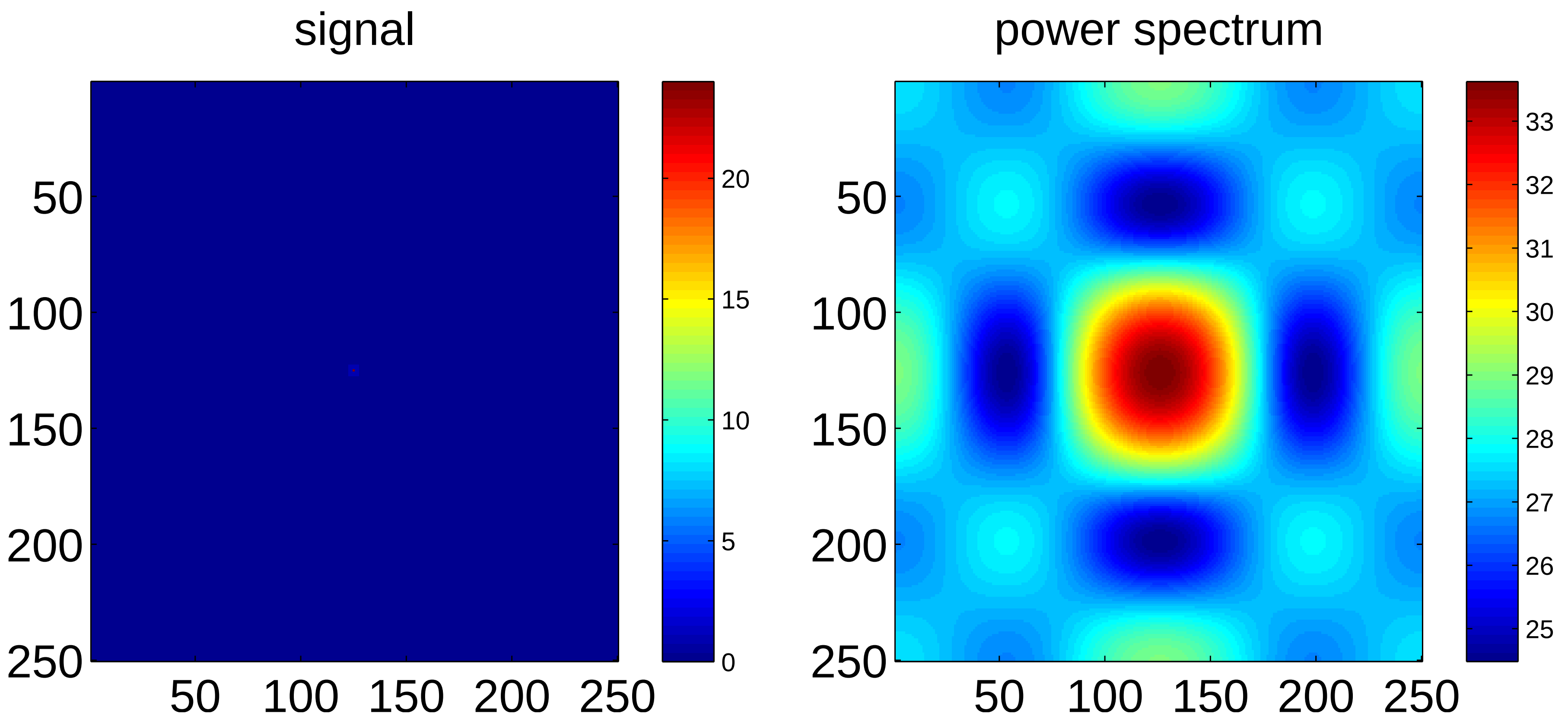
$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

5×5

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 24 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

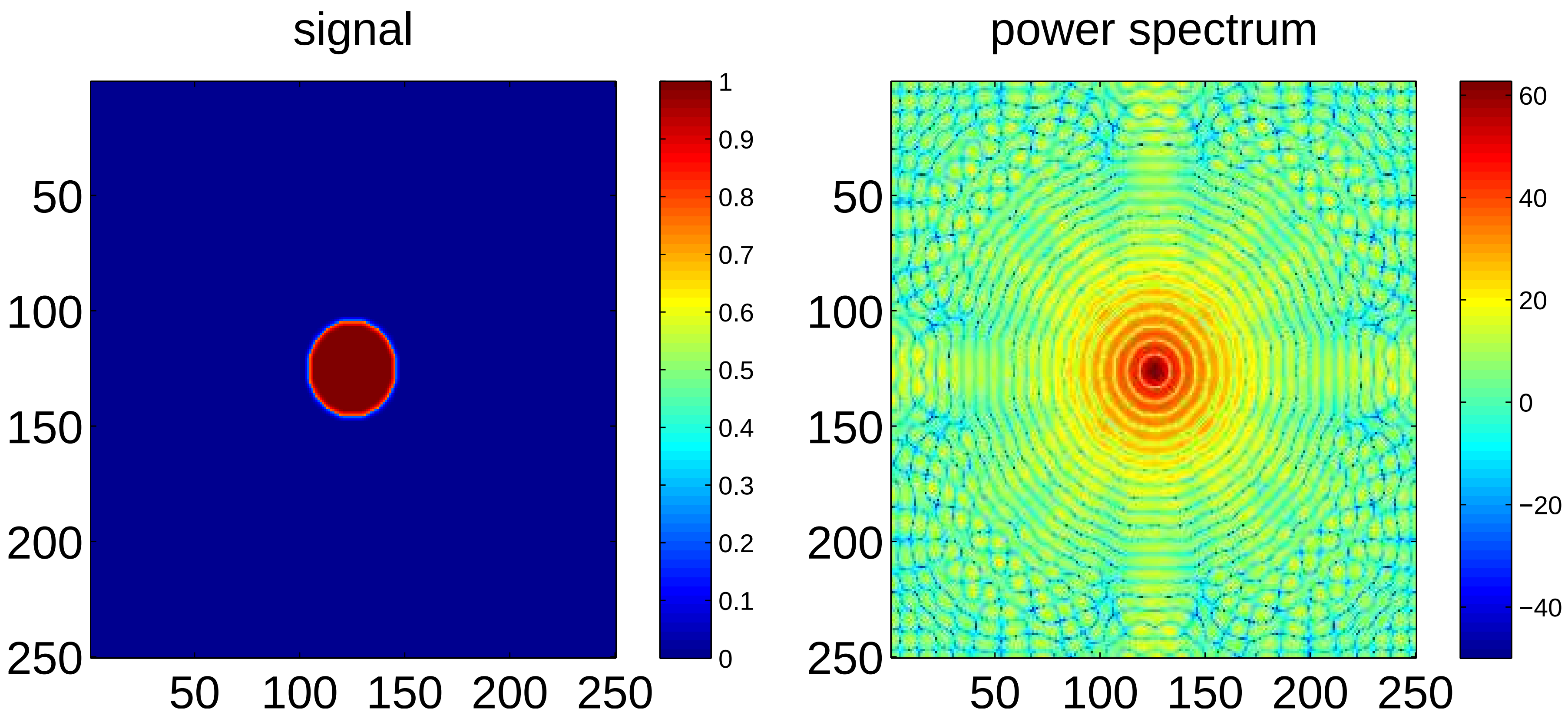
Filters in 2D

5×5 Laplacian, and its frequency response



Filters in 2D

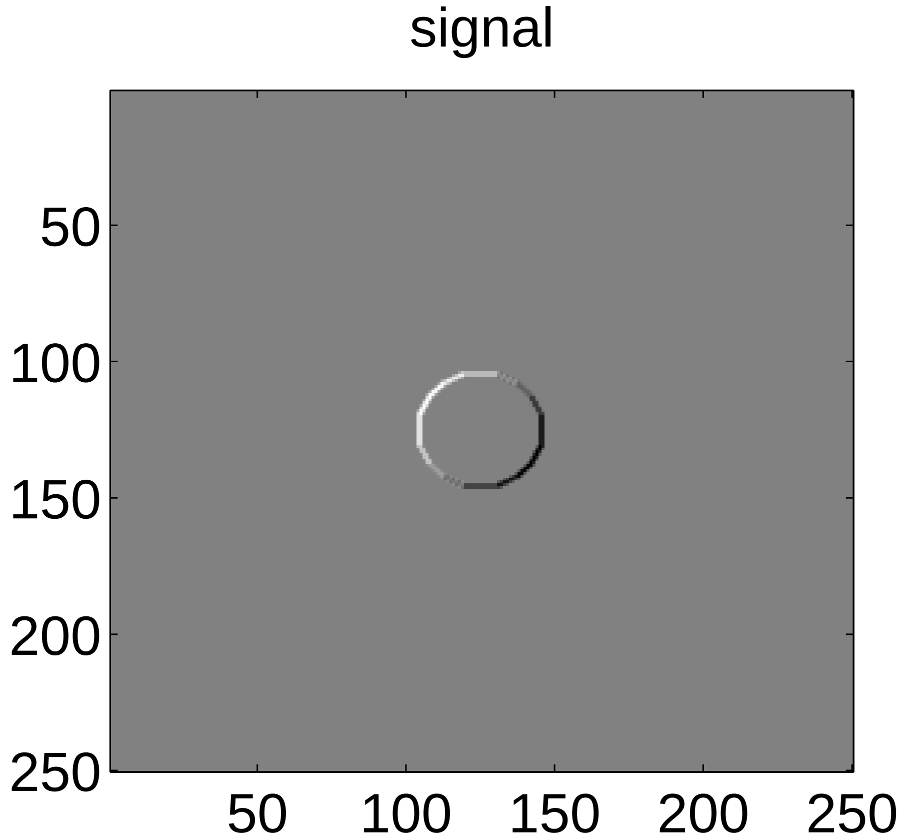
5×5 Laplacian applied to image



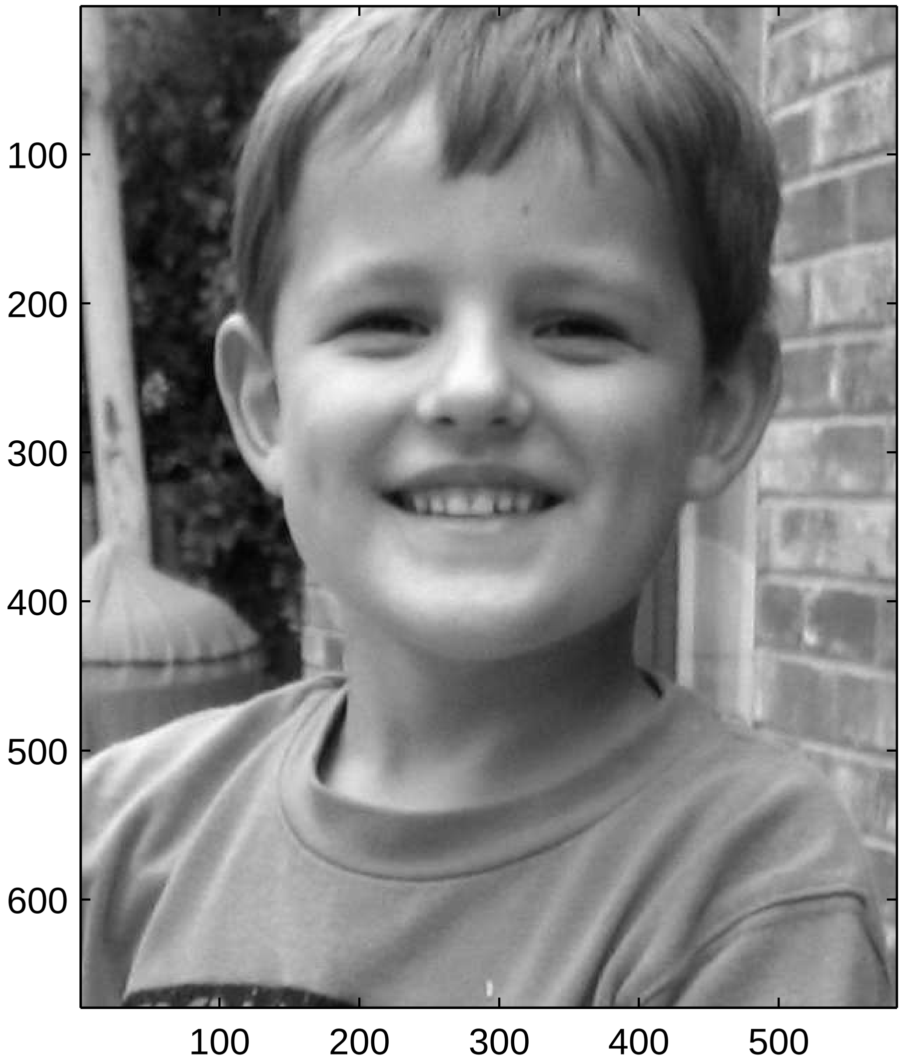
Embossing an image

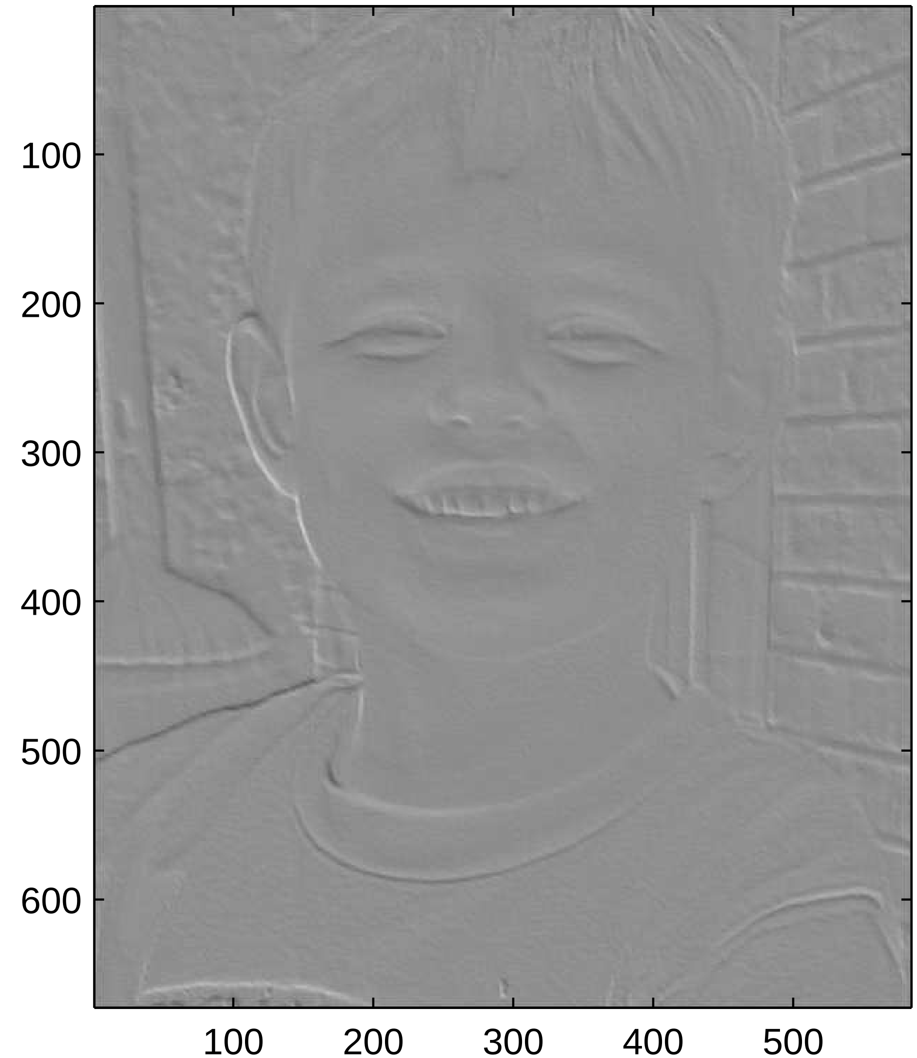
Fancy effects from simple filters, e.g. take $\theta = \pi/6$ and

$$\text{image} = M_{\text{sobel}}^{\text{horizontal}} \cos(\theta) + M_{\text{sobel}}^{\text{vertical}} \sin(\theta)$$



Embossing an image





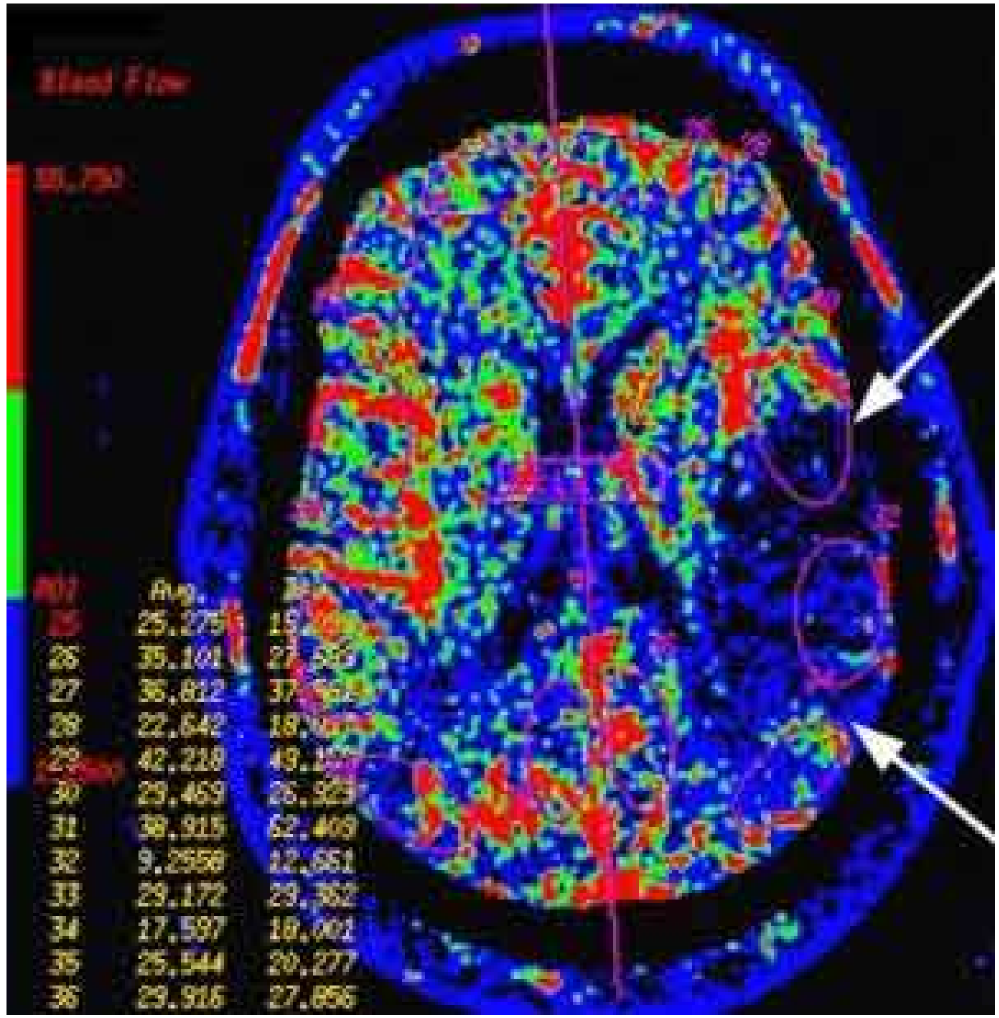
Tomography

One of the more fantastic achievements of signal processing is in the field of medical imaging where tomography is used to see inside a living body. The underlying techniques rely on transforms, and have wide applications in many other areas.

Applications: tomography

- in many problems we **can't** directly observe object
- we observe indirect measurements
  - seismology (determining size/shape of ore body)
  - oceanography (acoustic observations to get water temp)
  - archaeology (e.g. finding remains with GPR)
  - medical imaging (CT, MRI, PET scans)
  - manufacturing (process monitoring)
  - networks (traffic matrix estimation)
- inverse problem to take indirect measurements and infer true state from these
- tomography (from 'cuts' = tomo-)

Computed Aided Tomography

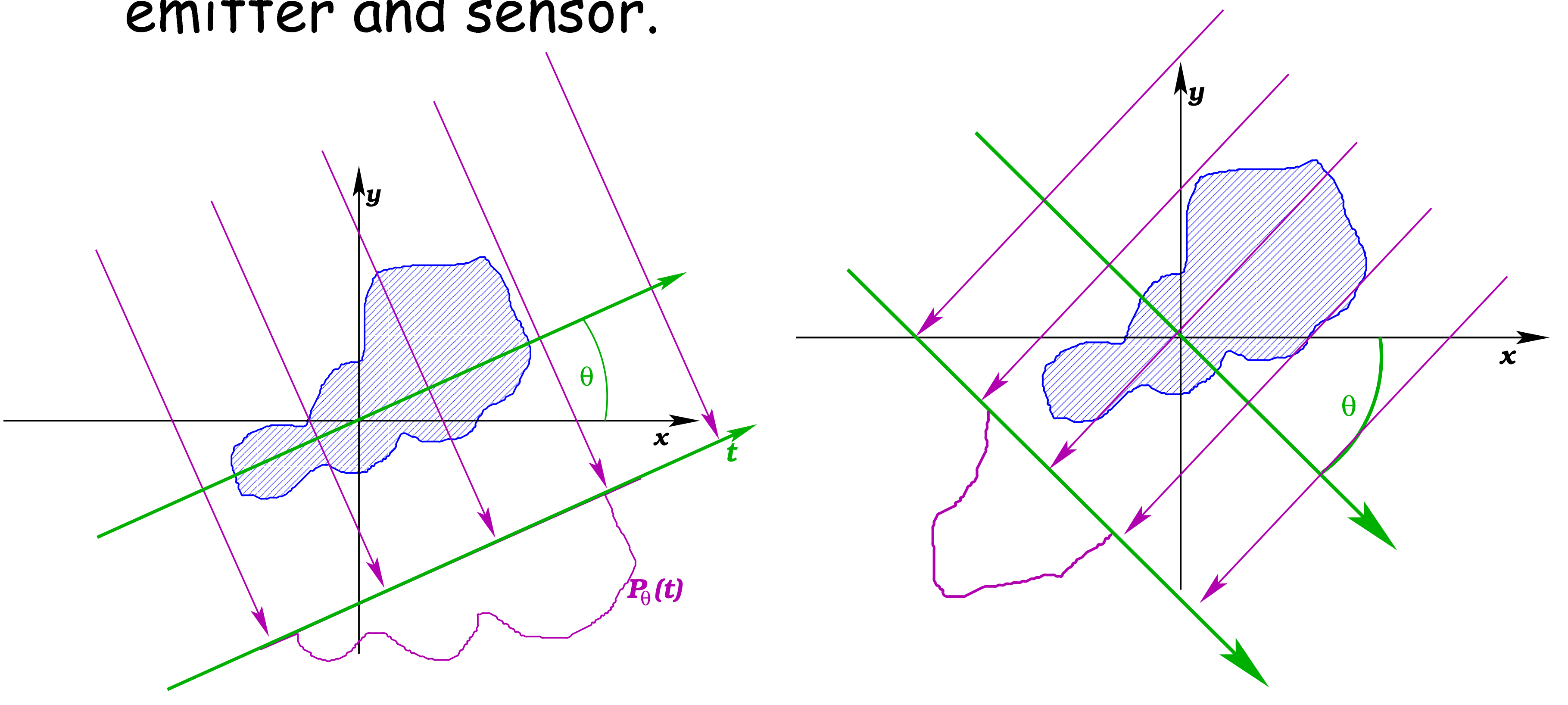


http://www.radiologyinfo.org/photocat/photos.cfm?image=hd-ct-perfusion.jpg
&&subcategory=Head&&stop=9

Computed Aided Tomography

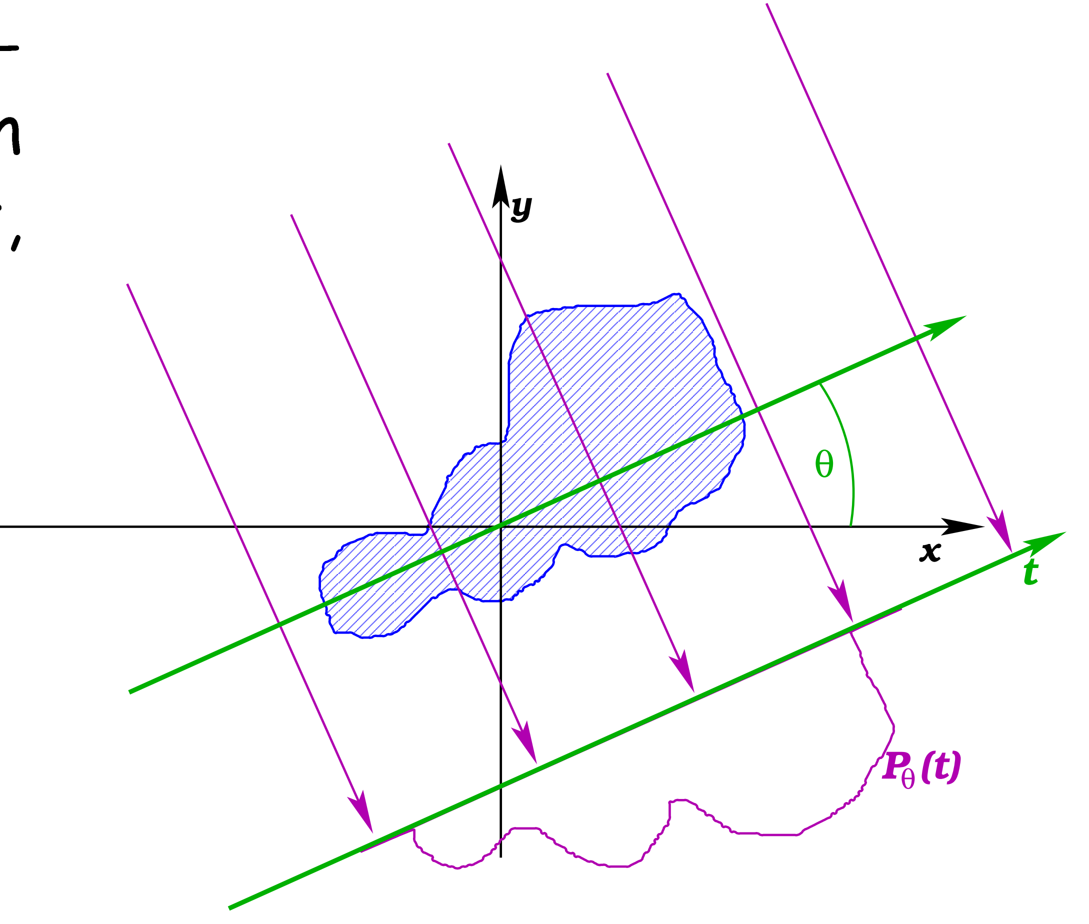
Simple view

- take a series of observations from different angles
- at each angle, measure density of material between emitter and sensor.



Computed Aided Tomography

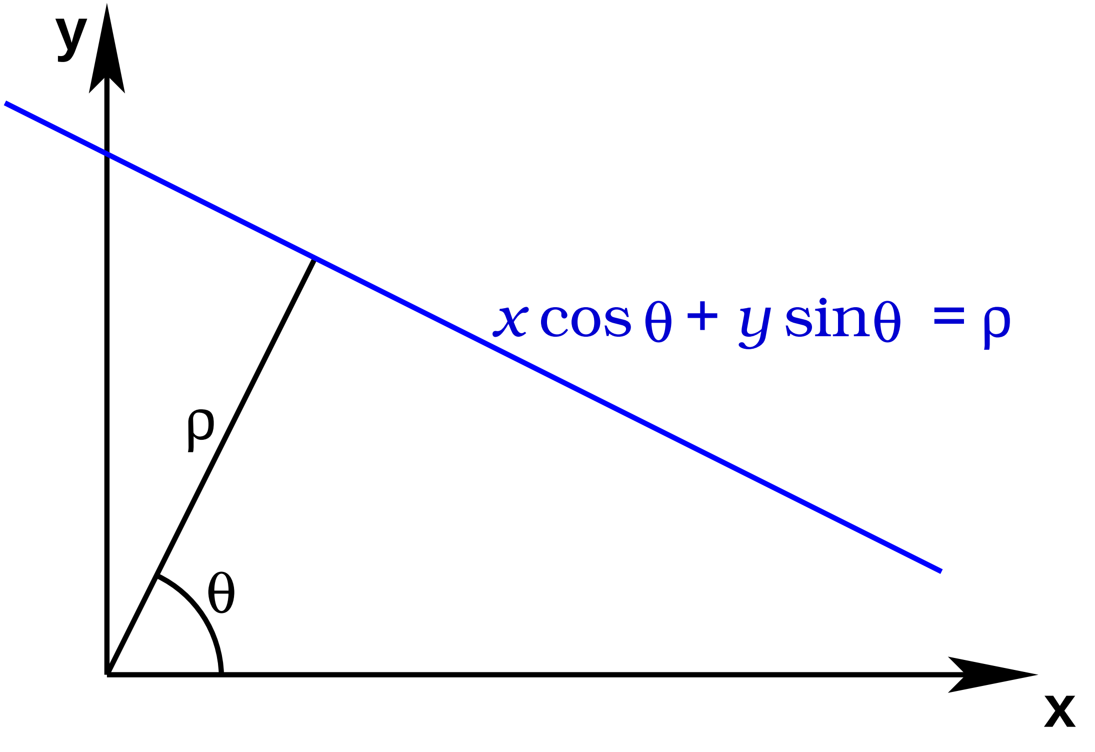
We can compute the projection $P_\theta(t)$ using the Radon transform of the object, e.g.



$$P_\theta(t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(t - x\cos\theta - y\sin\theta)\,dx\,dy$$

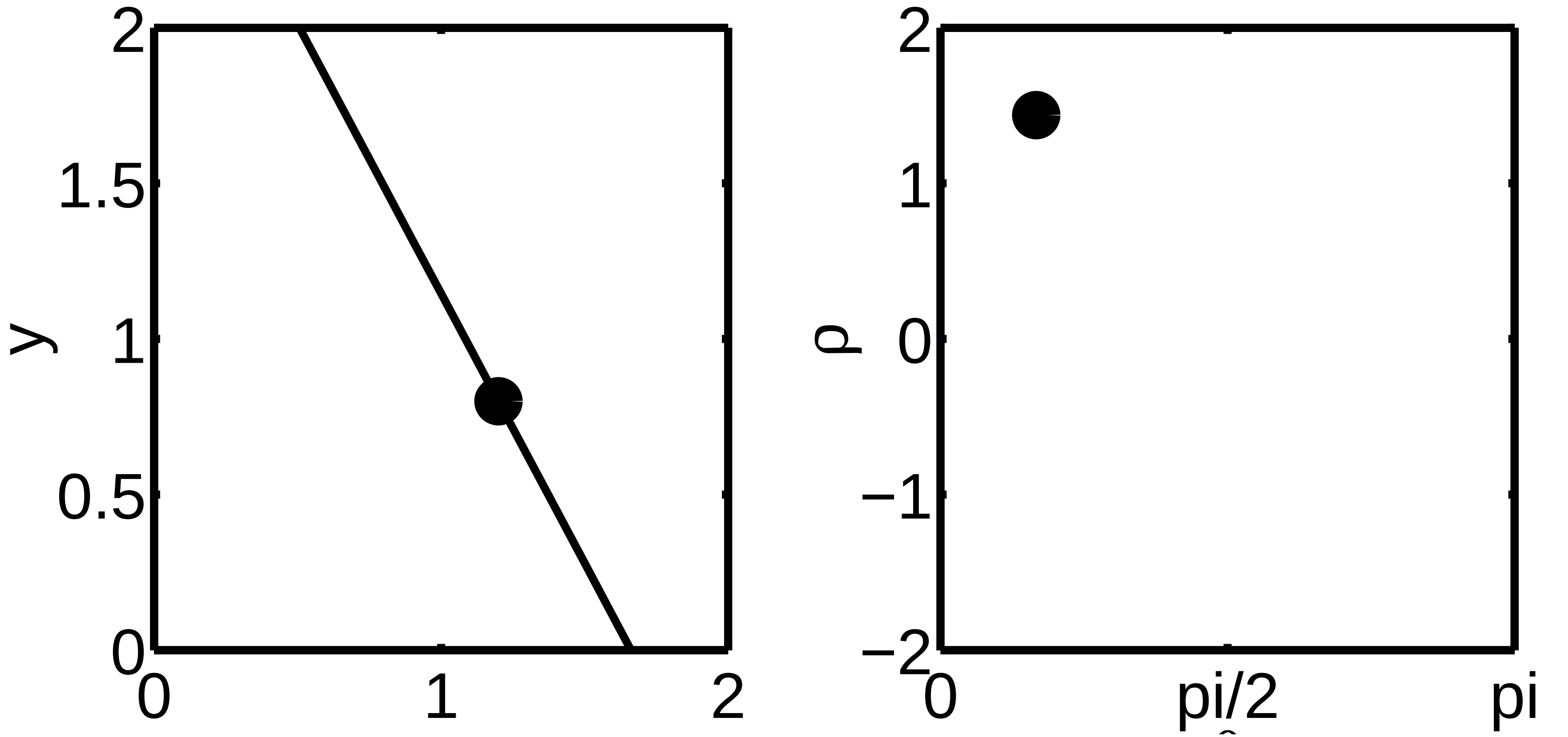
An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



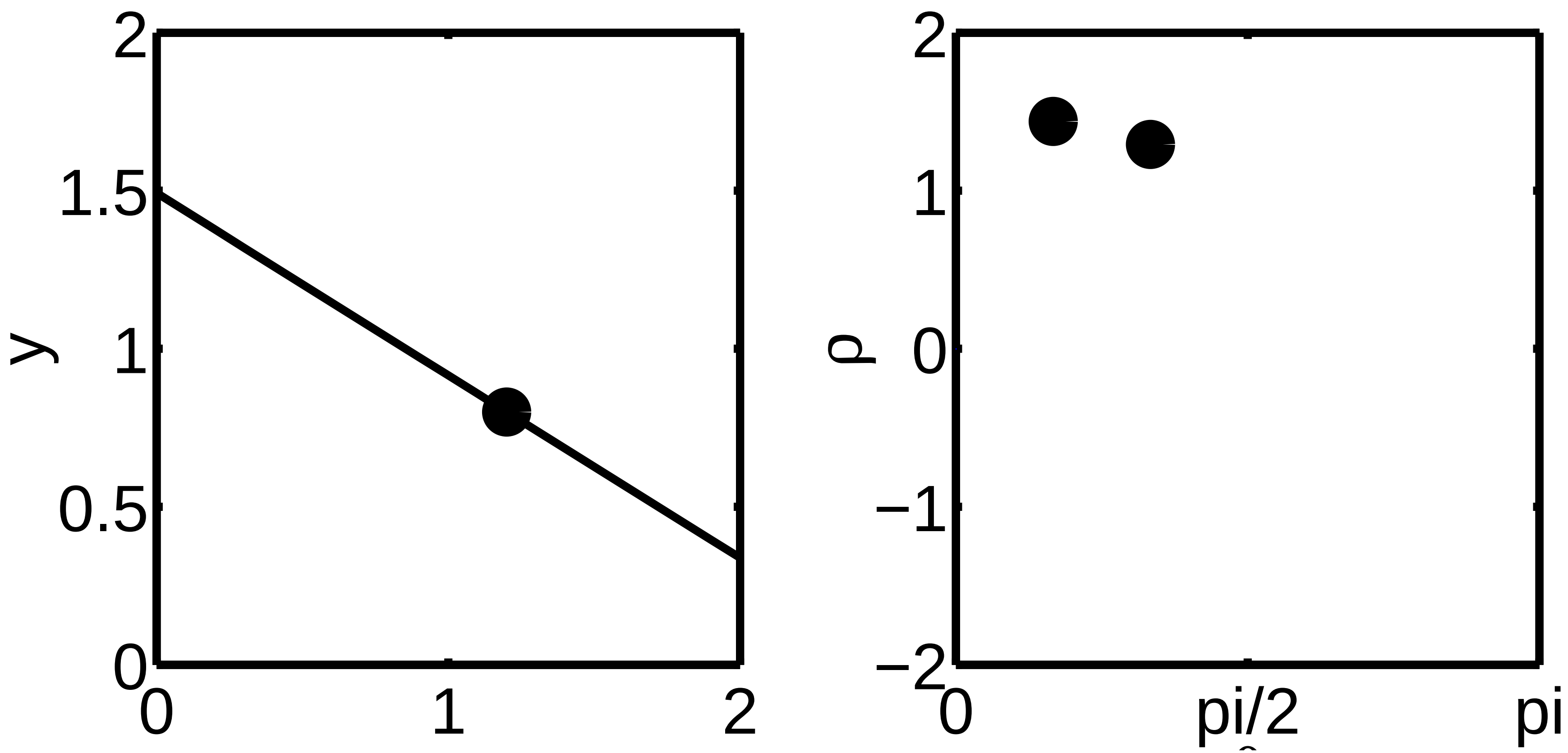
An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



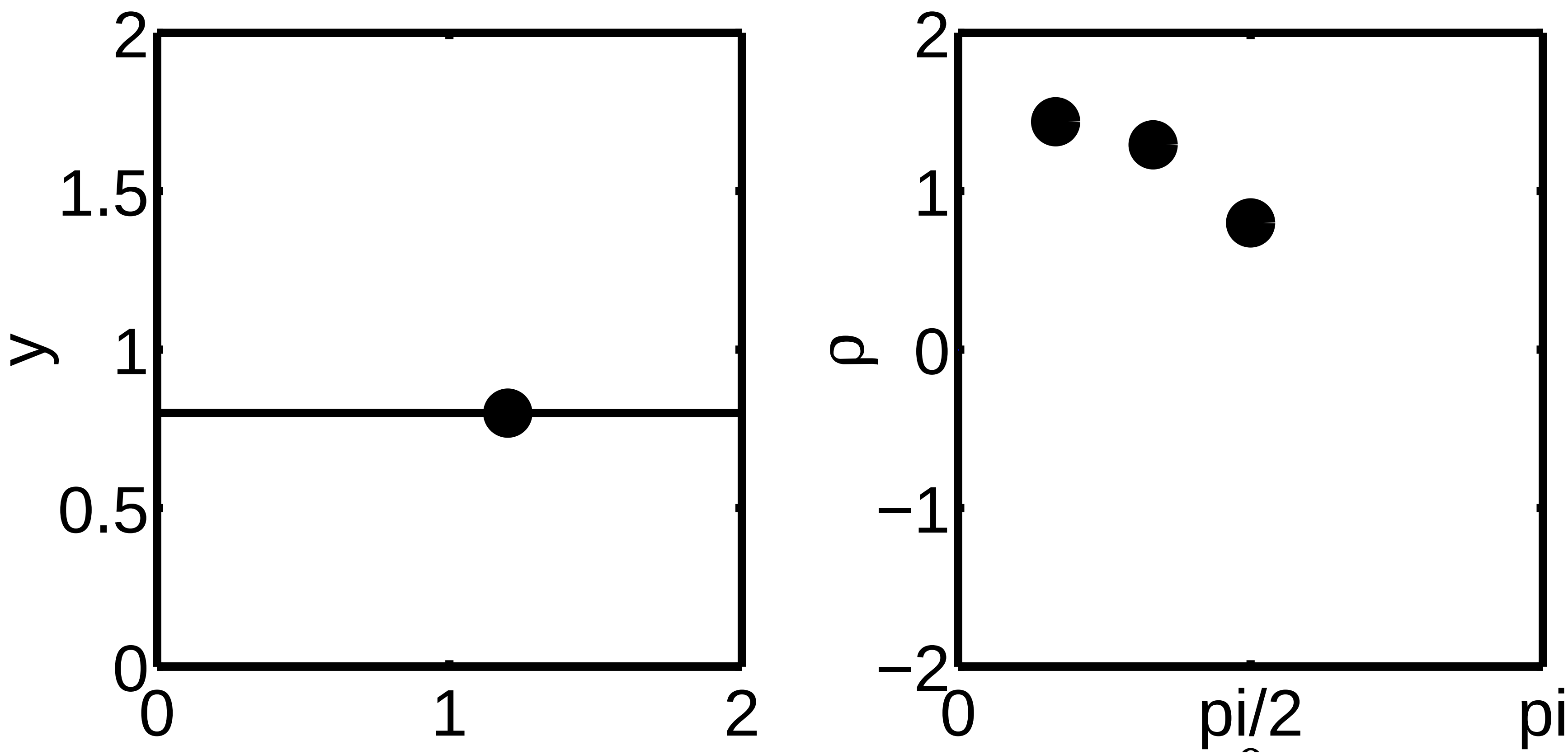
An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



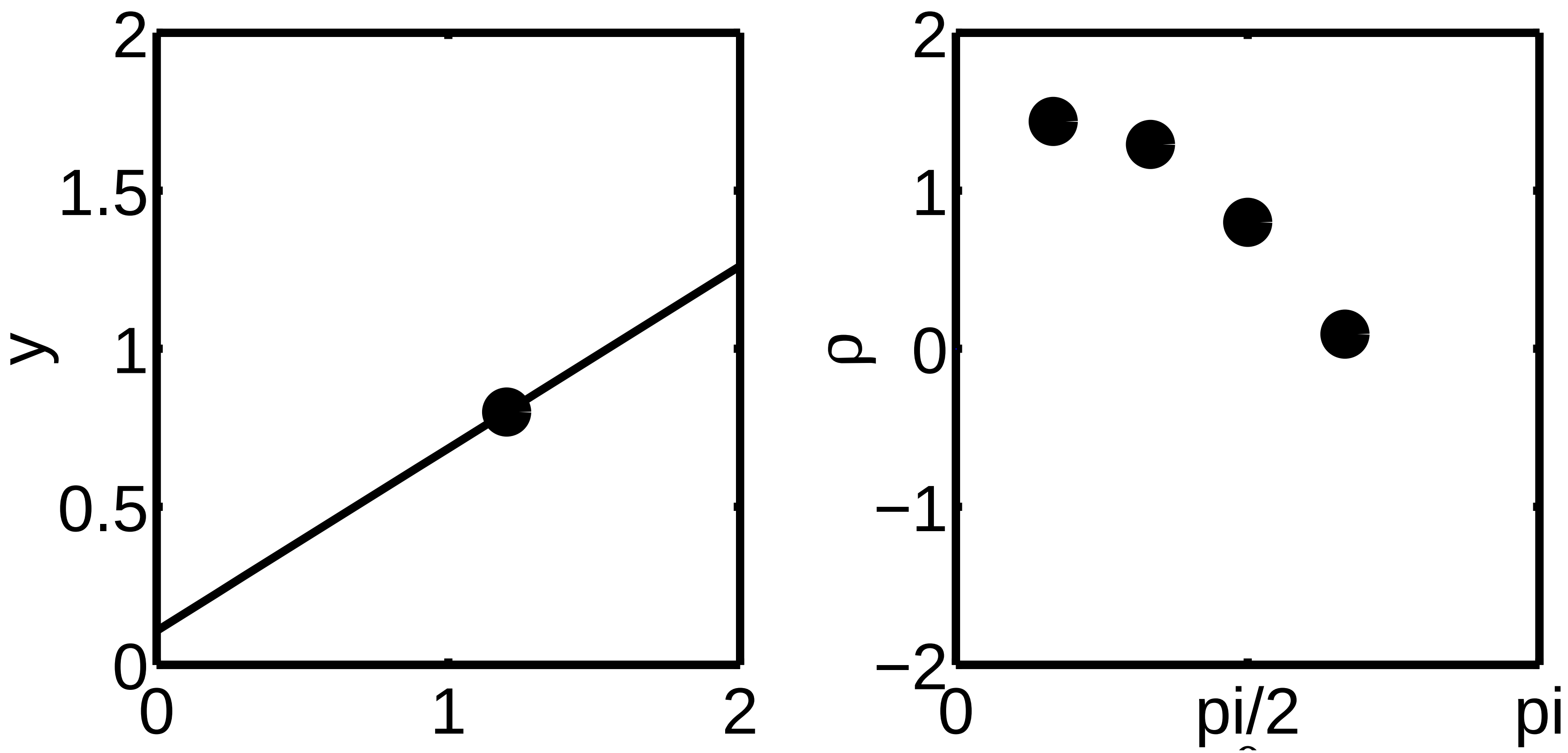
An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



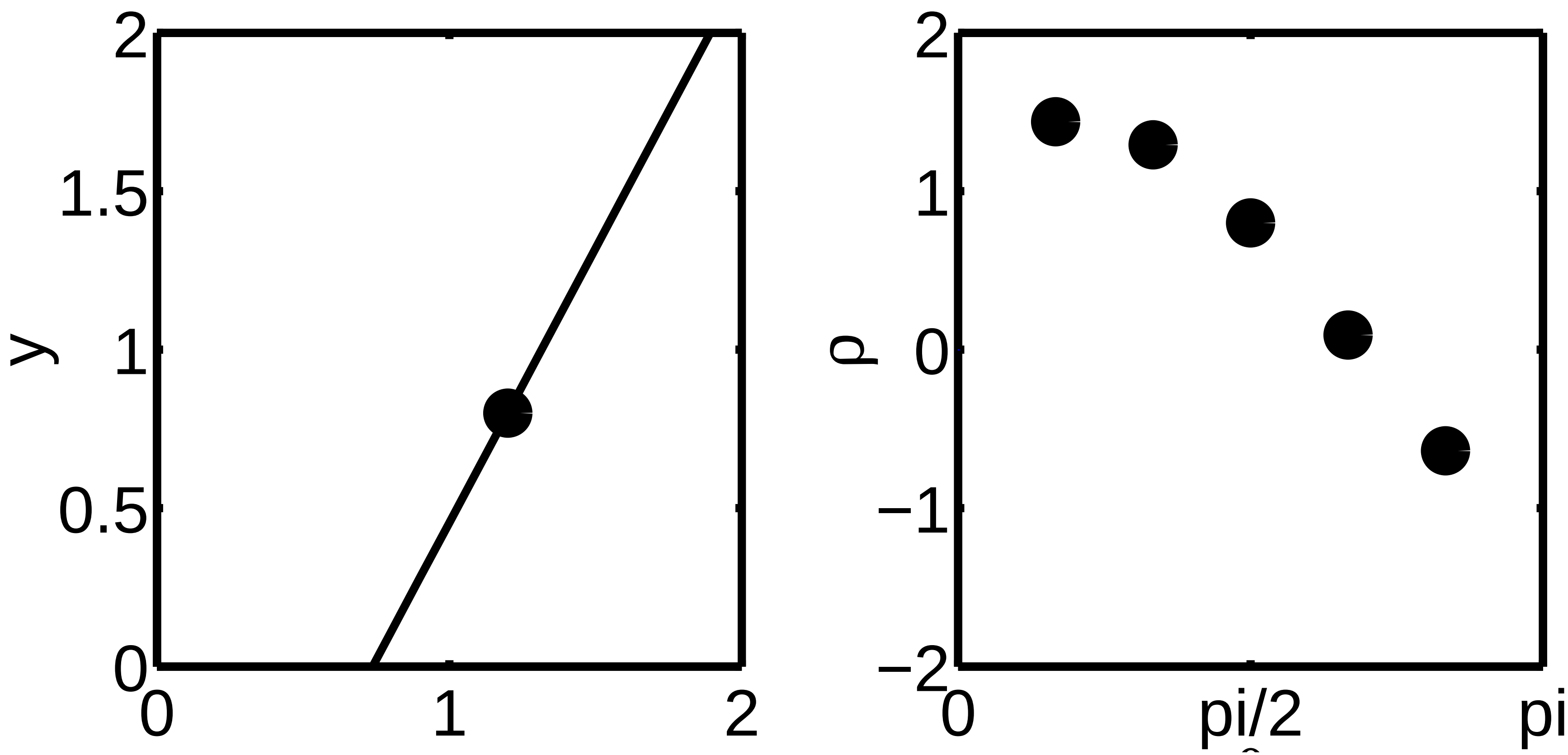
An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



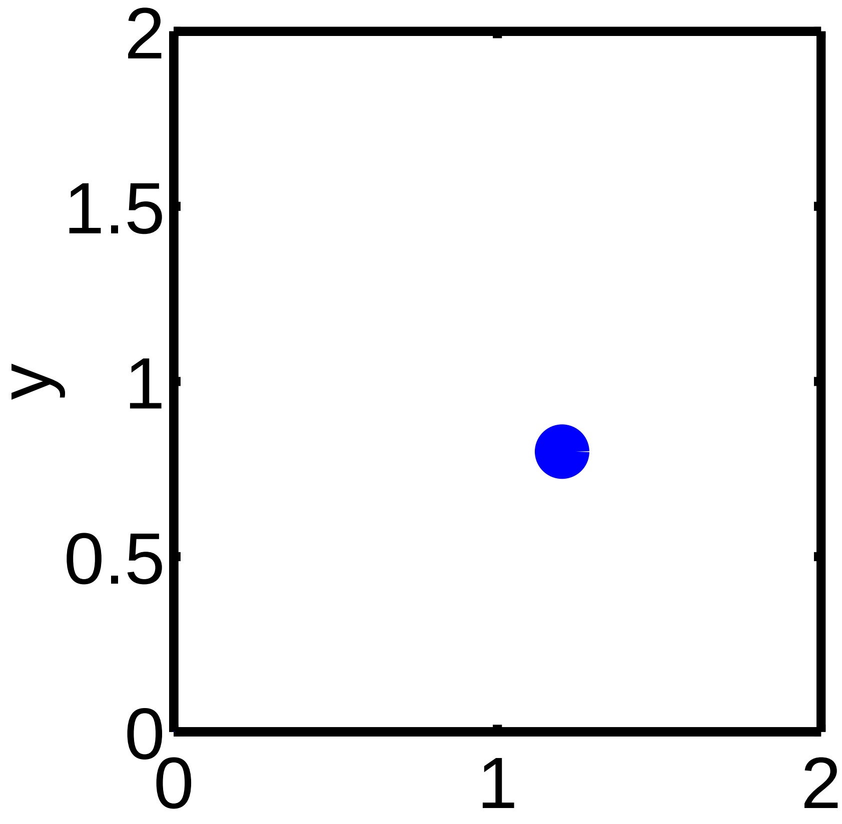
An example: Radon transform

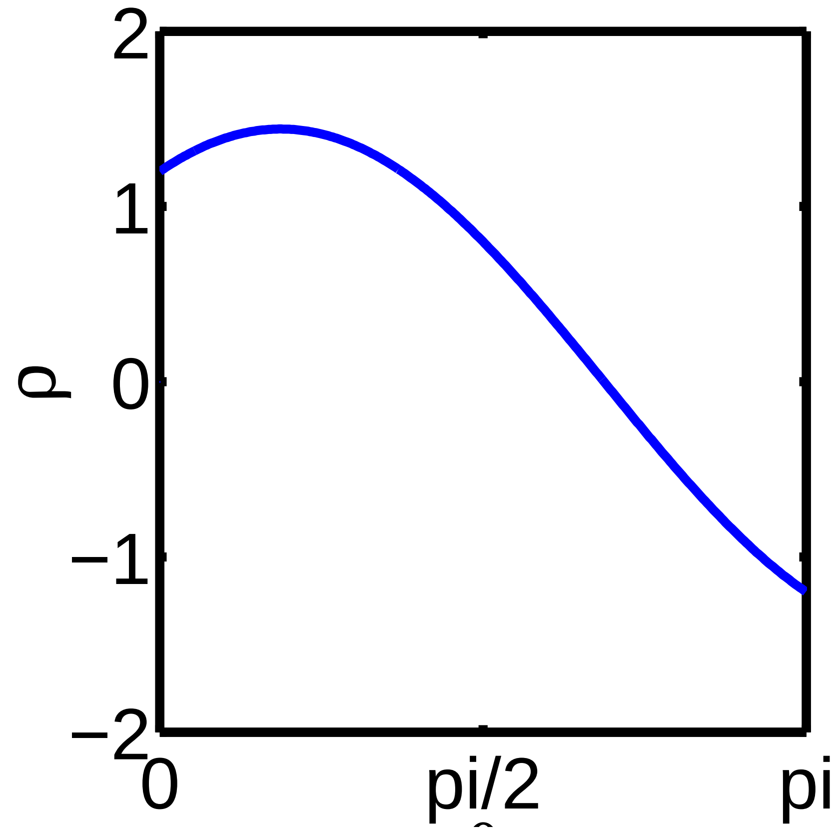
$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



An example: Radon transform

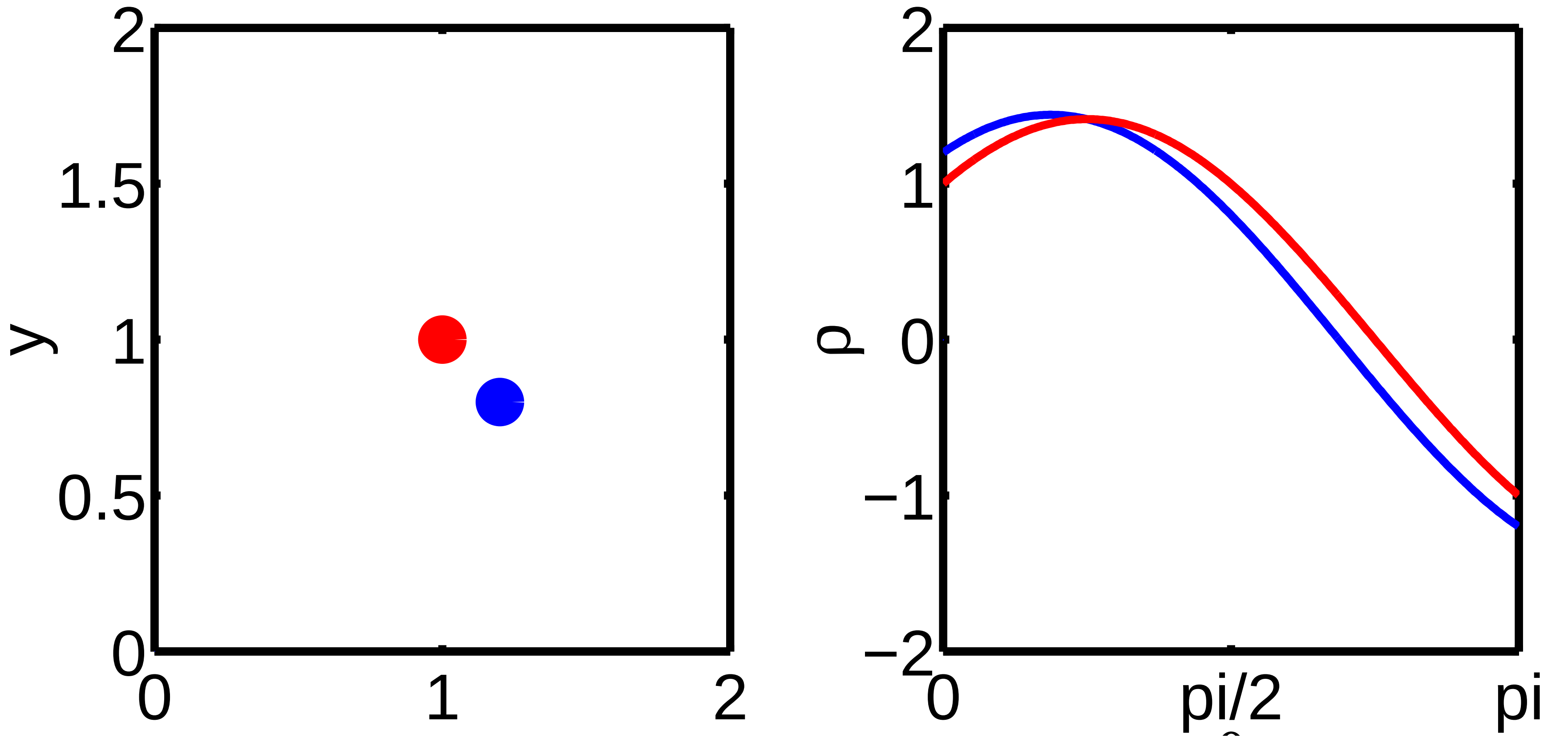
$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$





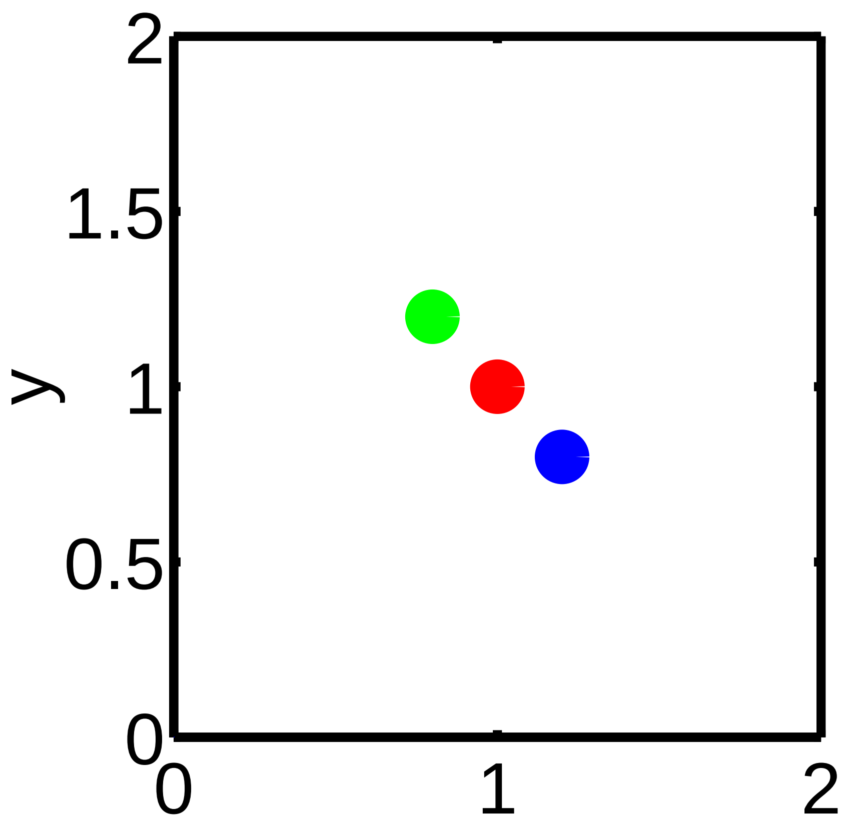
An example: Radon transform

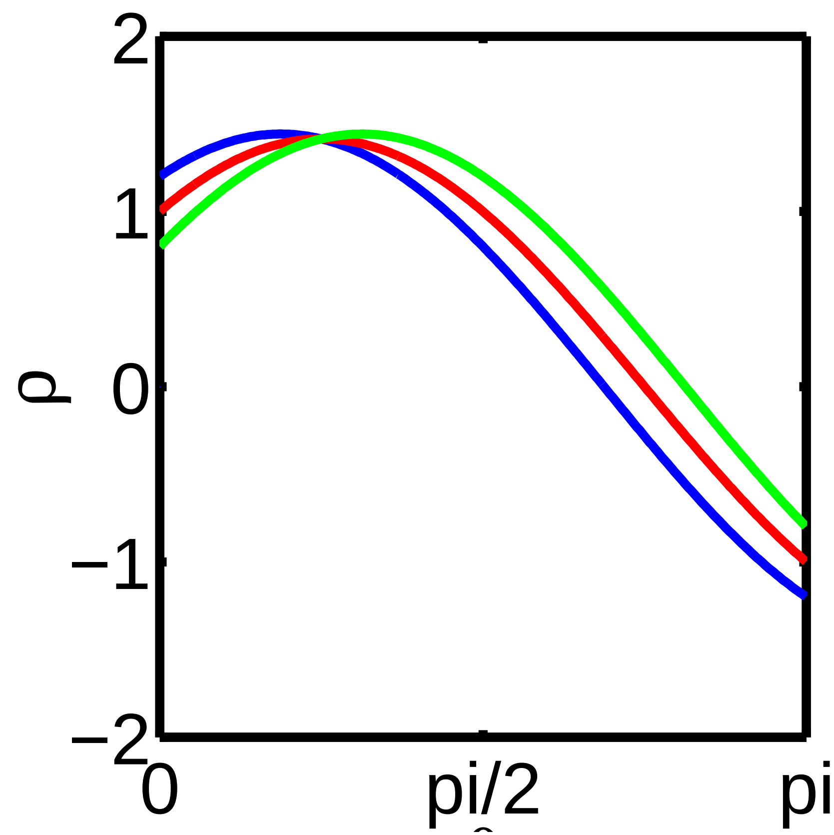
$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



An example: Radon transform

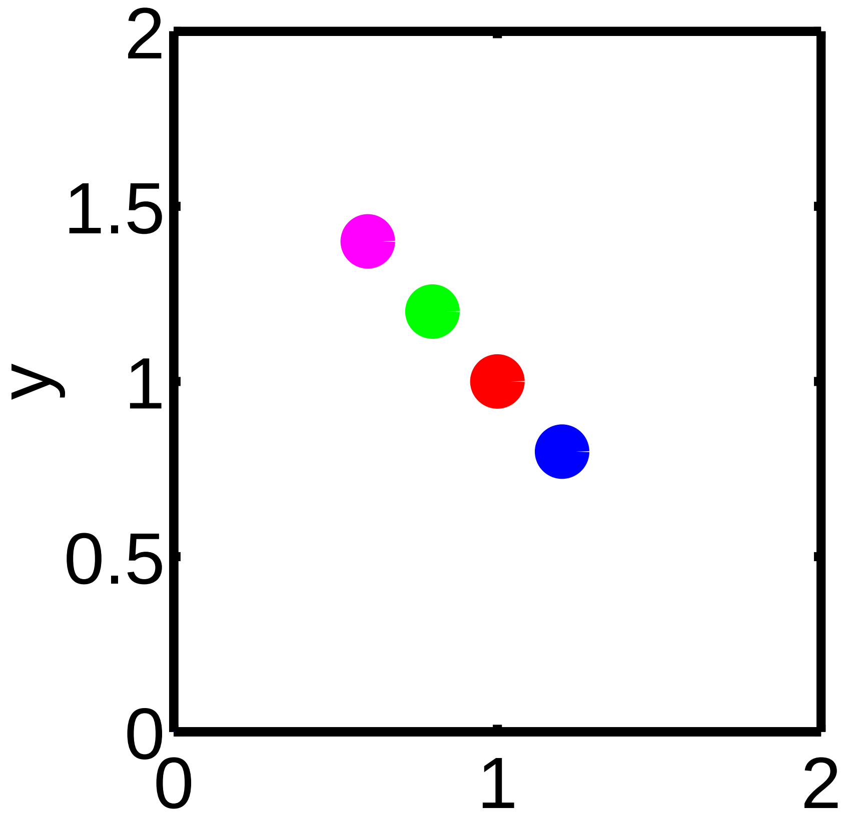
$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$

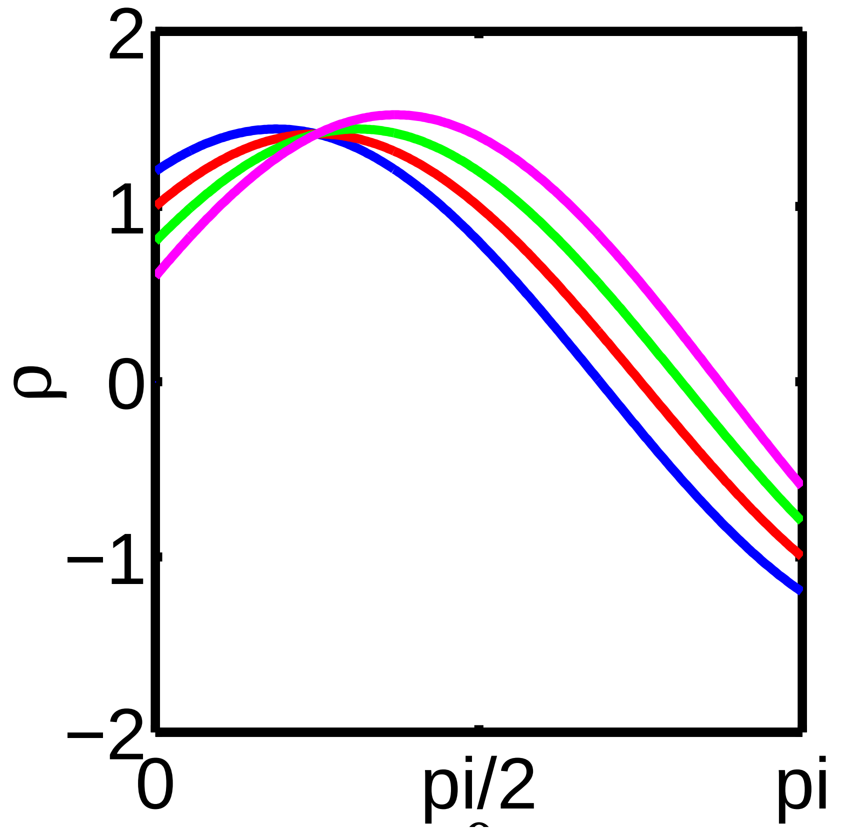




An example: Radon transform

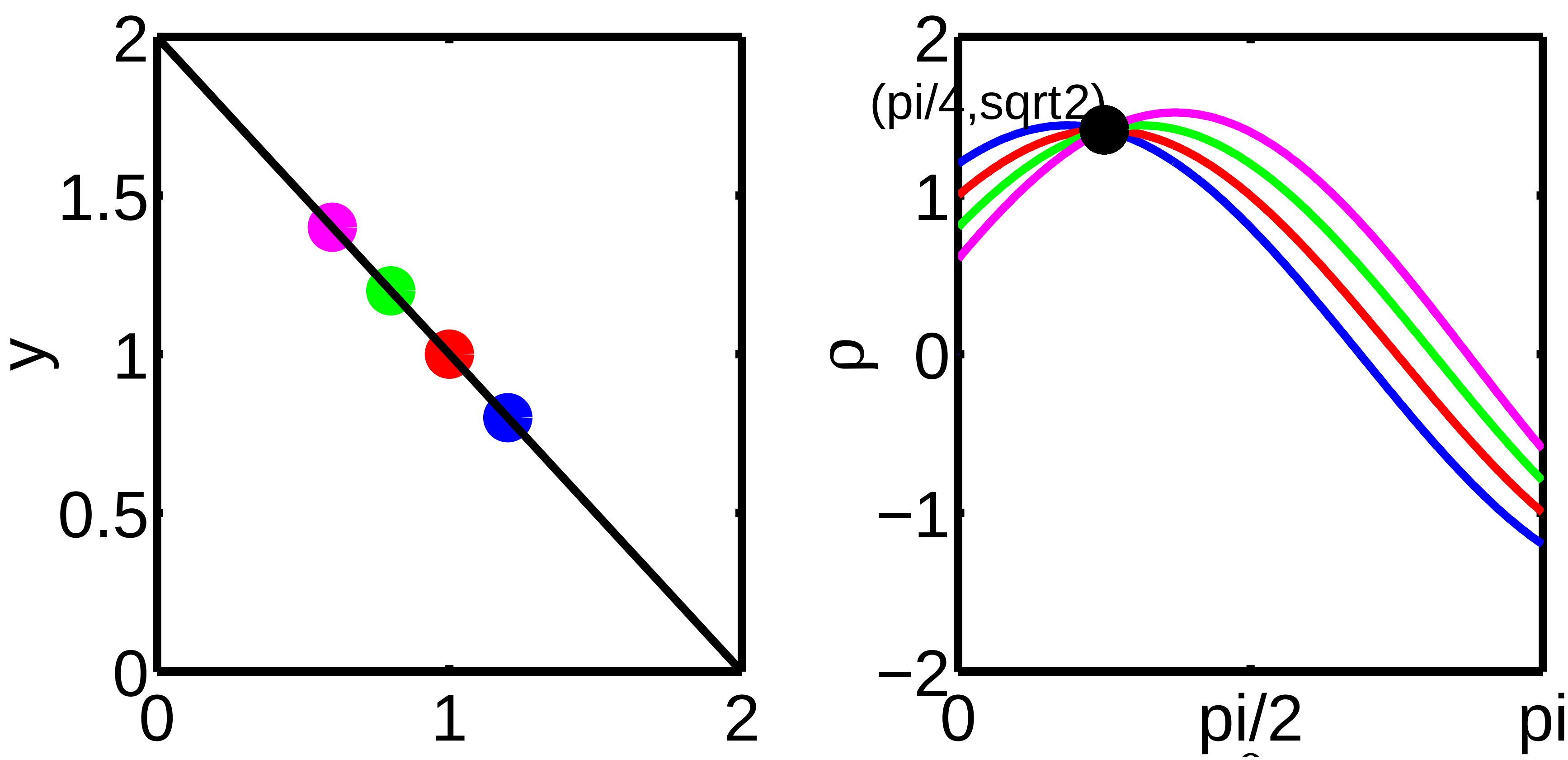
$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



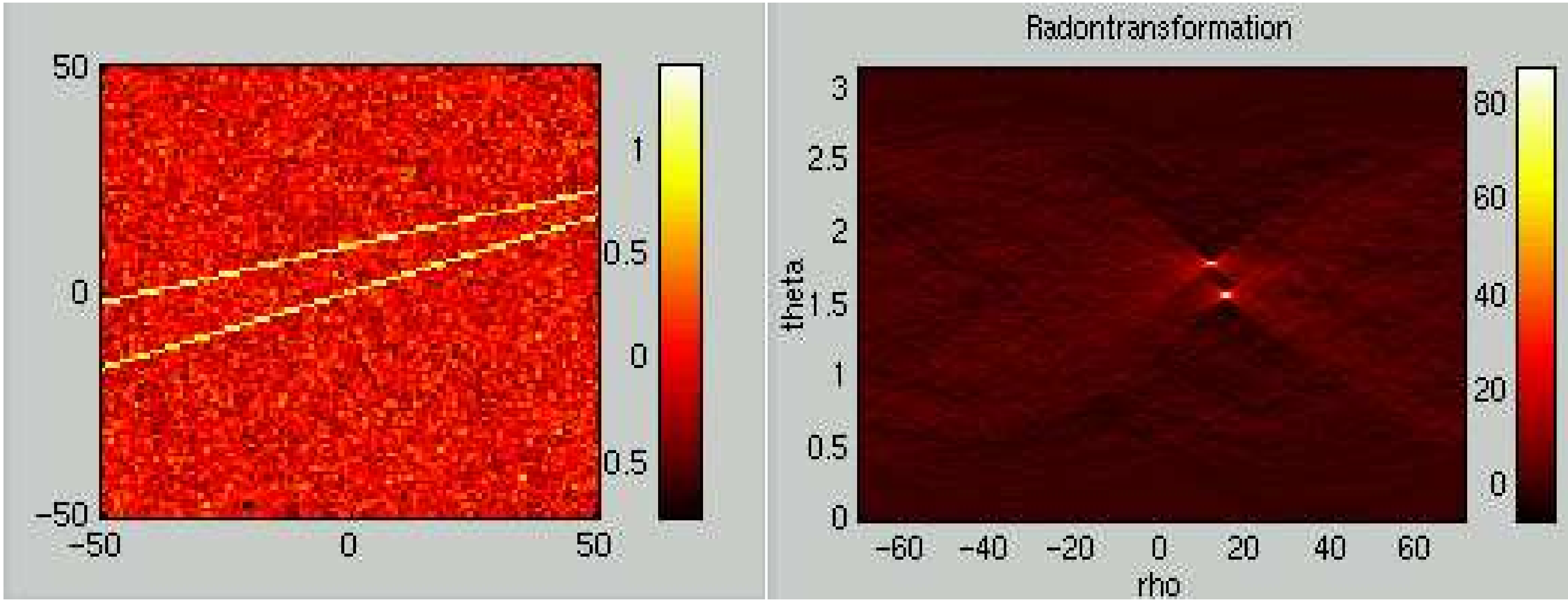


An example: Radon transform

$$F(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,dx\,dy$$



An example: Radon transform



http://eivind.imm.dtu.dk/staff/ptoft/Radon/Radon.html

Fourier transform of the cut

We can compute the FT of a slice

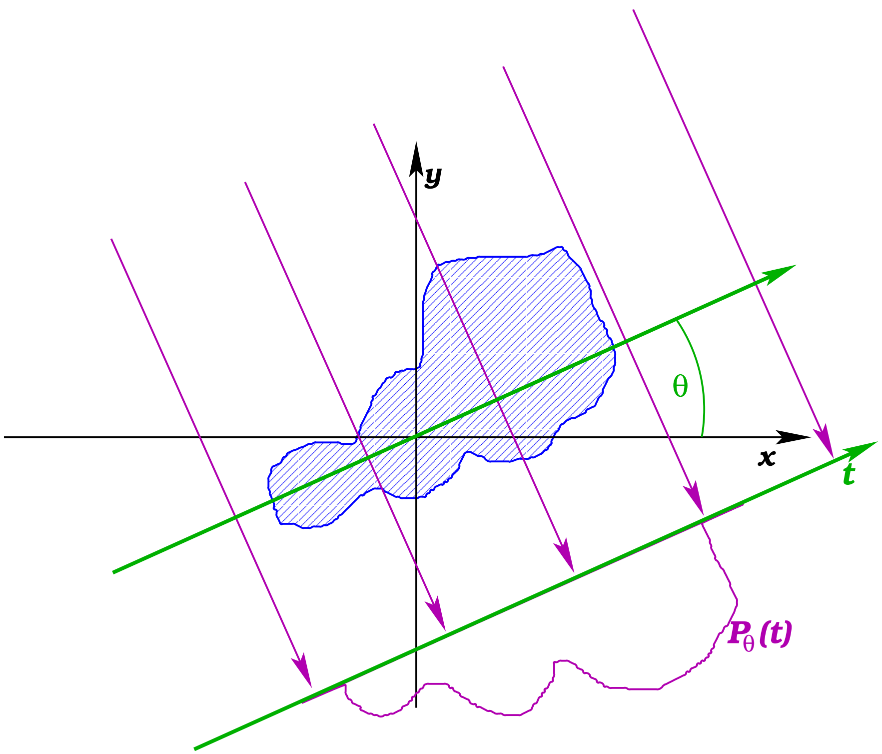
$$
\begin{aligned}
F_\theta(s) &= \int_{-\infty}^{\infty} P_\theta(t) e^{-i2\pi st}\, dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, \delta(t - x\cos\theta - y\sin\theta)\, dx\, dy\, e^{-i2\pi st}\, dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \int_{-\infty}^{\infty} \delta(t - x\cos\theta - y\sin\theta) e^{-i2\pi st}\, dt\, dx\, dy \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i2\pi s(x\cos\theta + y\sin\theta)}\, dx\, dy \\
&= F(s\cos\theta, s\sin\theta)
\end{aligned}
$$

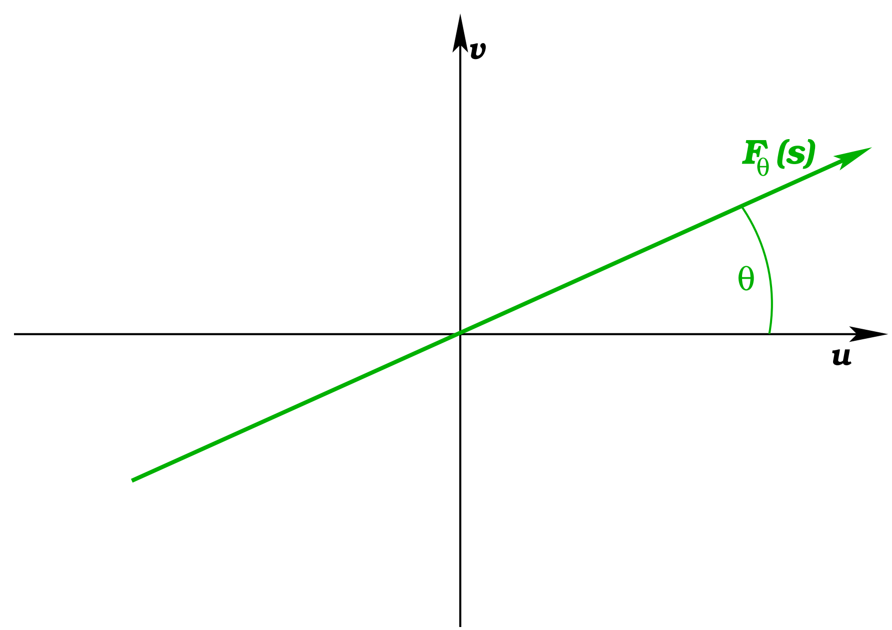
Fourier slice theorem

The Fourier transform of a projection (slice) through the object $F_\theta(s)$

$$F_\theta(s) = F(s\cos\theta, s\sin\theta)$$

where $F(u,v)$ is the 2D Fourier transform of the object.





Notional algorithm

We can now draw a notional algorithm

- Radon transform (via slices)
- Fourier transform to get slices of FT
- inverse Fourier transform to get back to object

But this fails to take into account

- discrete measurements
- finite number of projections

Back-projection filtering algorithm

Write inverse FT polar coordinates:

$$
\begin{aligned}
u &= \omega \cos\theta \\
v &= \omega \sin\theta
\end{aligned}
$$

So $du\,dv = \omega\, d\omega\, d\theta$, and

$$
\begin{aligned}
f(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{i2\pi(ux+vy)}\, du\, dv \\
&= \int_{0}^{2\pi}\int_{0}^{\infty} F(\omega,\theta) e^{i2\pi\omega(x\cos\theta + y\sin\theta)}\, \omega\, d\omega\, d\theta \\
&= \int_{0}^{\pi}\int_{0}^{\infty} F(\omega,\theta) e^{i2\pi\omega(x\cos\theta + y\sin\theta)}\, \omega\, d\omega\, d\theta \\
&\quad + \int_{0}^{\pi}\int_{0}^{\infty} F(\omega,\theta+\pi) e^{i2\pi\omega(x\cos(\theta+\pi) + y\sin(\theta+\pi))}\, \omega\, d\omega\, d\theta
\end{aligned}
$$

Back-projection filtering algorithm

From symmetry $F(\omega,\theta+\pi) = F(-\omega,\theta)$, so

$$\begin{aligned} f(x,y) &= \int_0^{\pi}\int_{-\infty}^{\infty} F(\omega,\theta) e^{i2\pi\omega(x\cos\theta+y\sin\theta)}\,|\omega|\,d\omega\,d\theta \\ &= \int_0^{\pi}\int_{-\infty}^{\infty} F_\theta(\omega) e^{i2\pi\omega(x\cos\theta+y\sin\theta)}\,|\omega|\,d\omega\,d\theta \\ &= \int_0^{\pi} Q_\theta(x\cos\theta+y\sin\theta)\,d\theta \end{aligned}$$

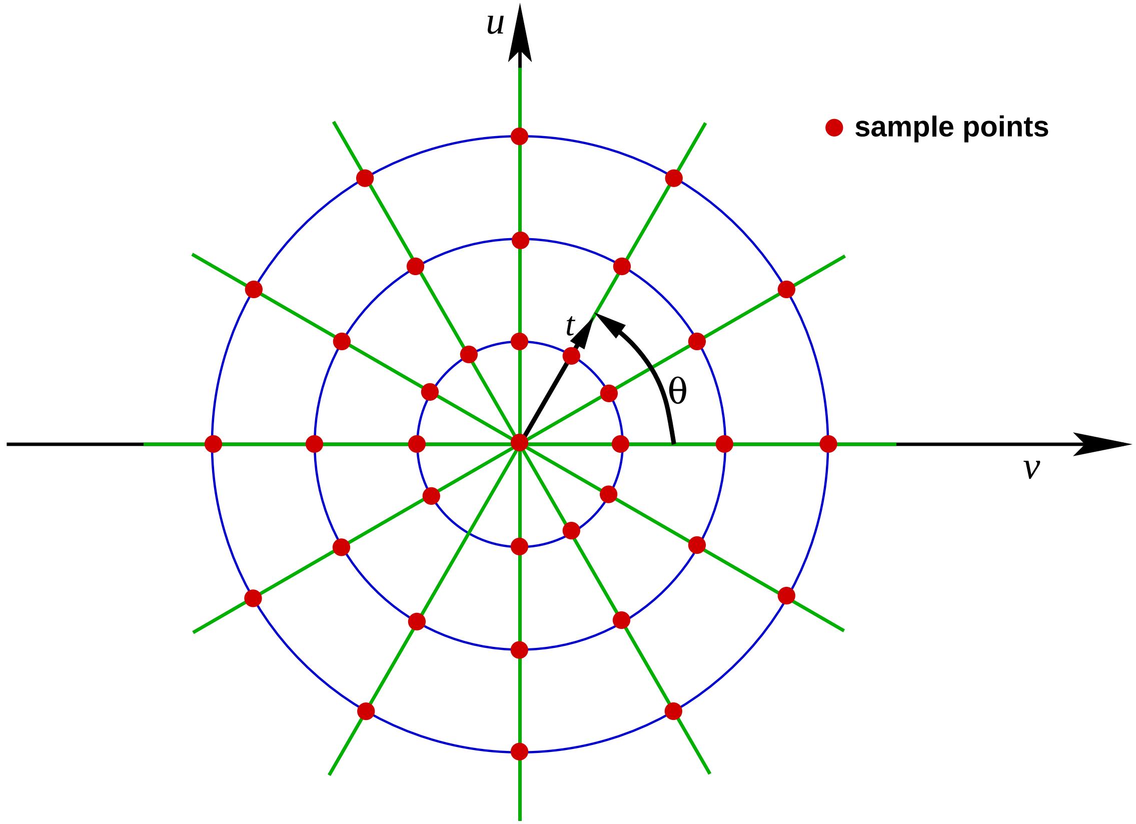
where

$$Q_\theta(t) = \int_{-\infty}^{\infty} F_\theta(\omega)|\omega| e^{i2\pi\omega t}\,d\omega$$

$Q_\theta(t)$ represents a filtered version of $P_\theta(t)$, with frequency response $|\omega|$.

Sampling of Fourier Domain

Assume that t and θ are sampled uniformly, the sampling of the Fourier domain looks like.

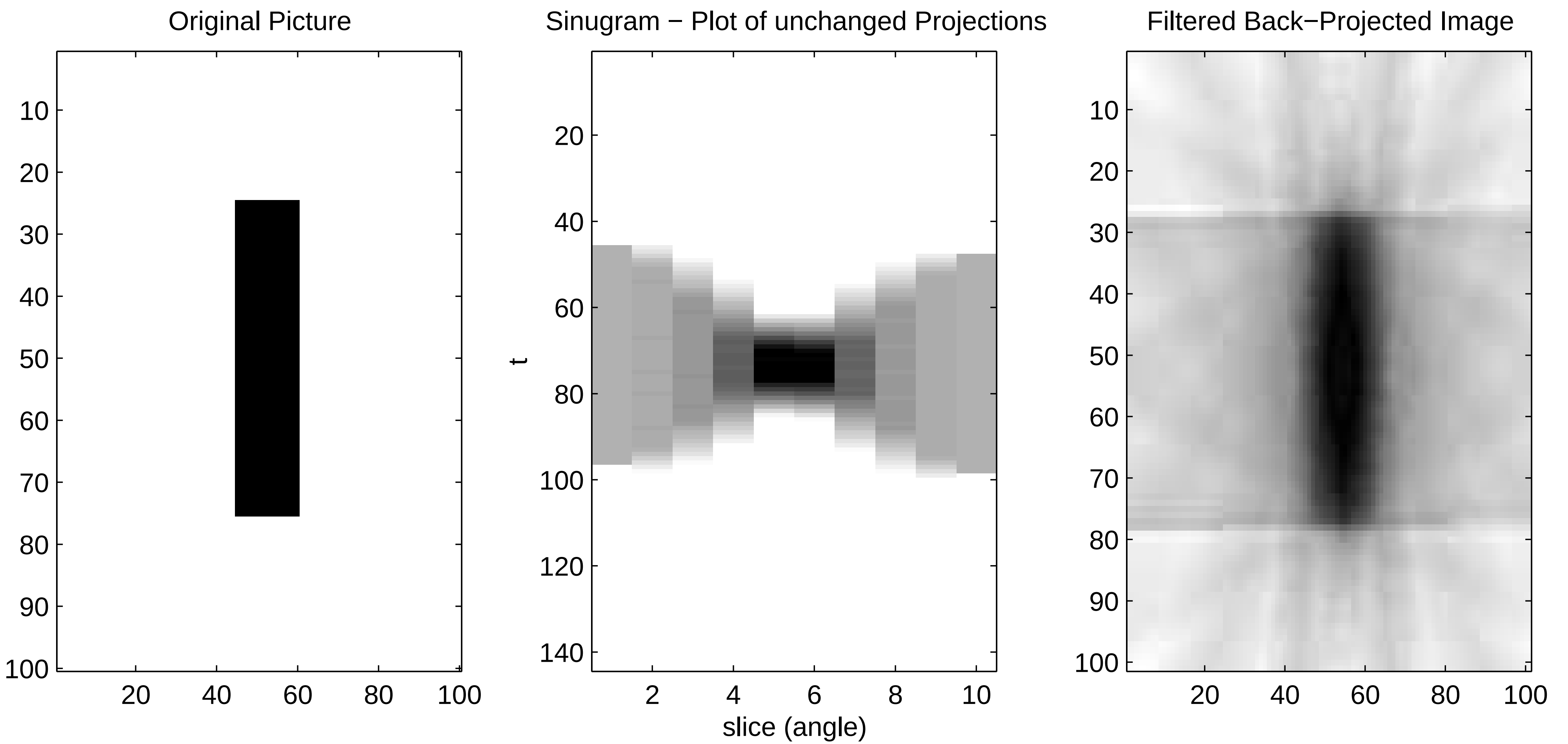


Samples are more dense around the center!

- filtering tries to counteract this

Examples

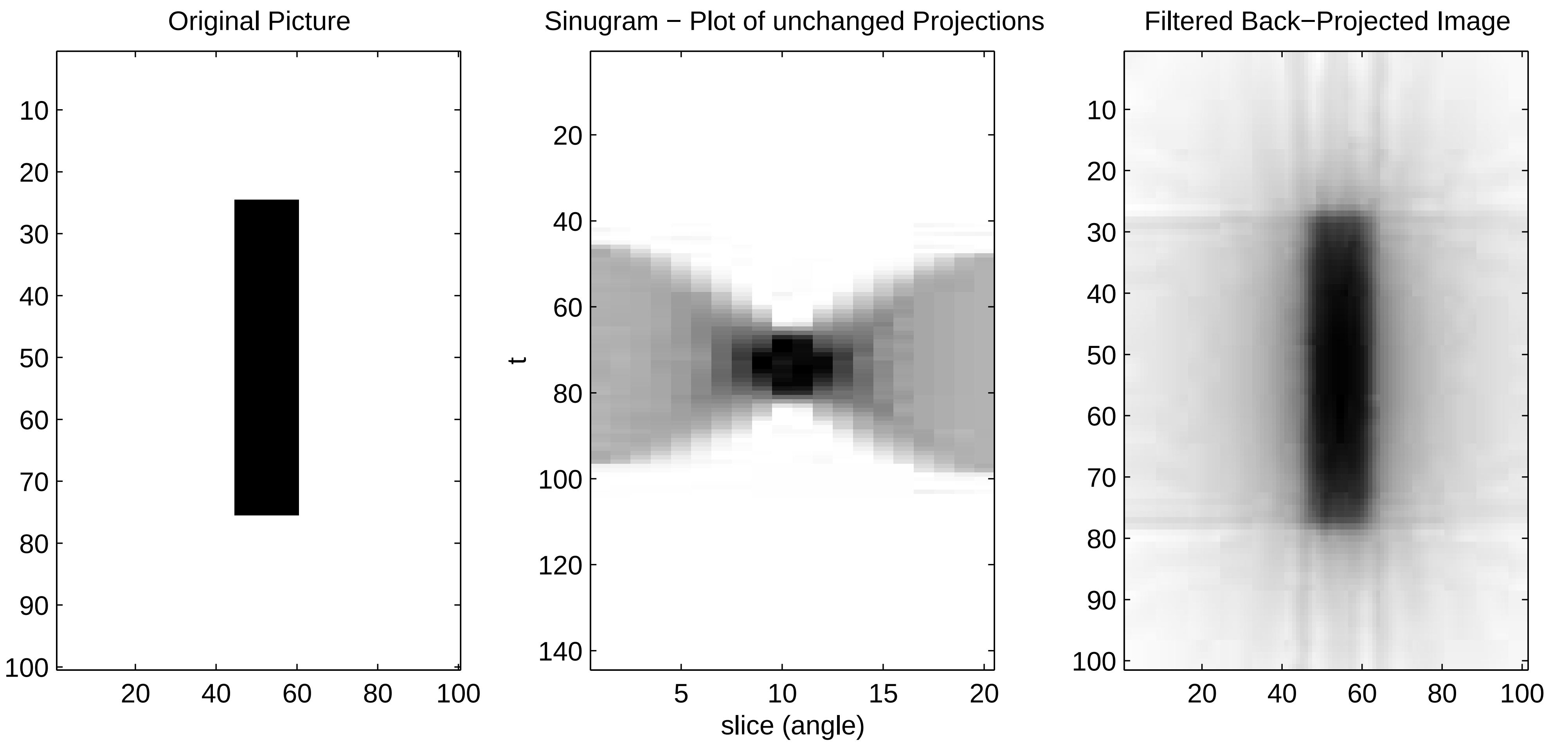
10 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

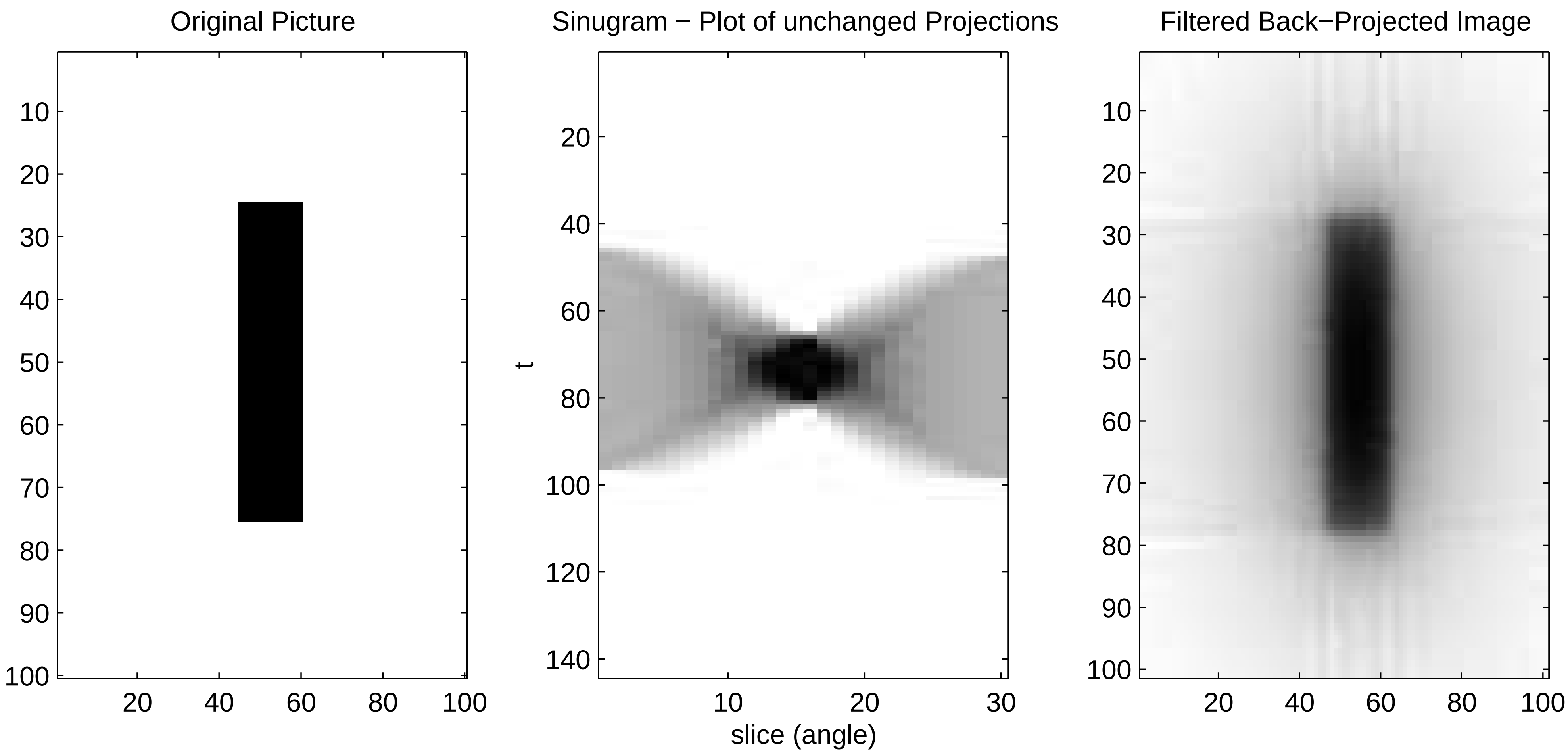
20 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

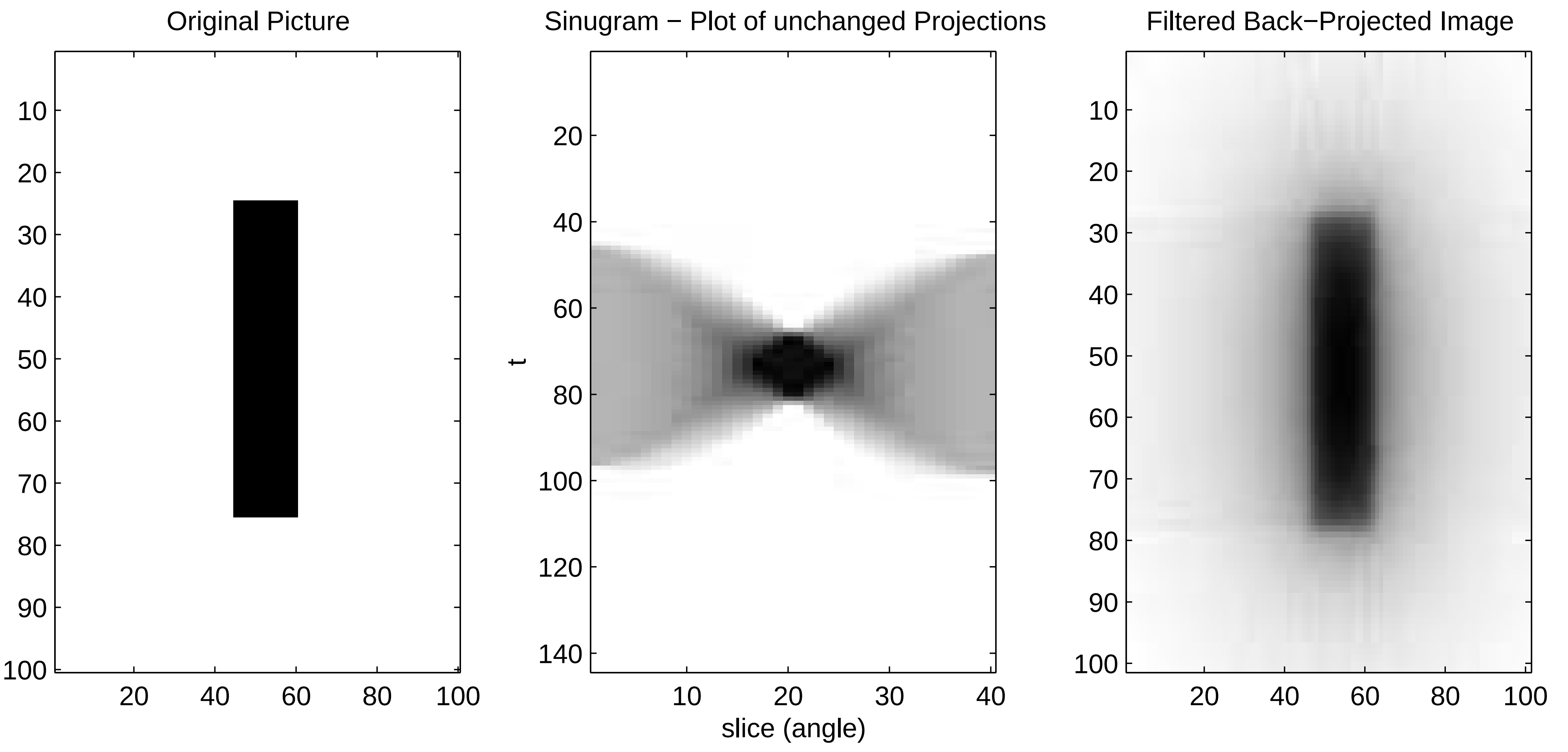
30 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

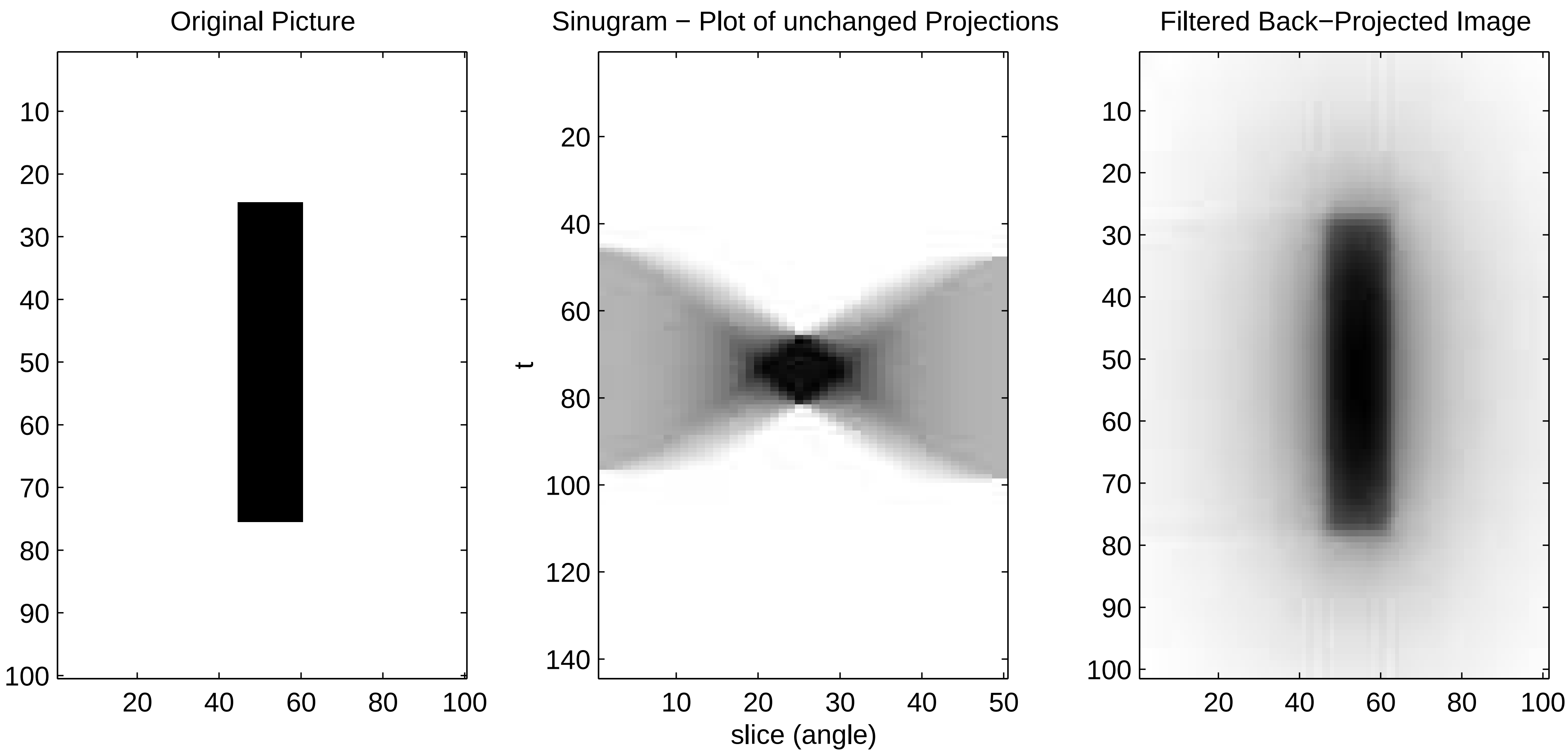
40 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

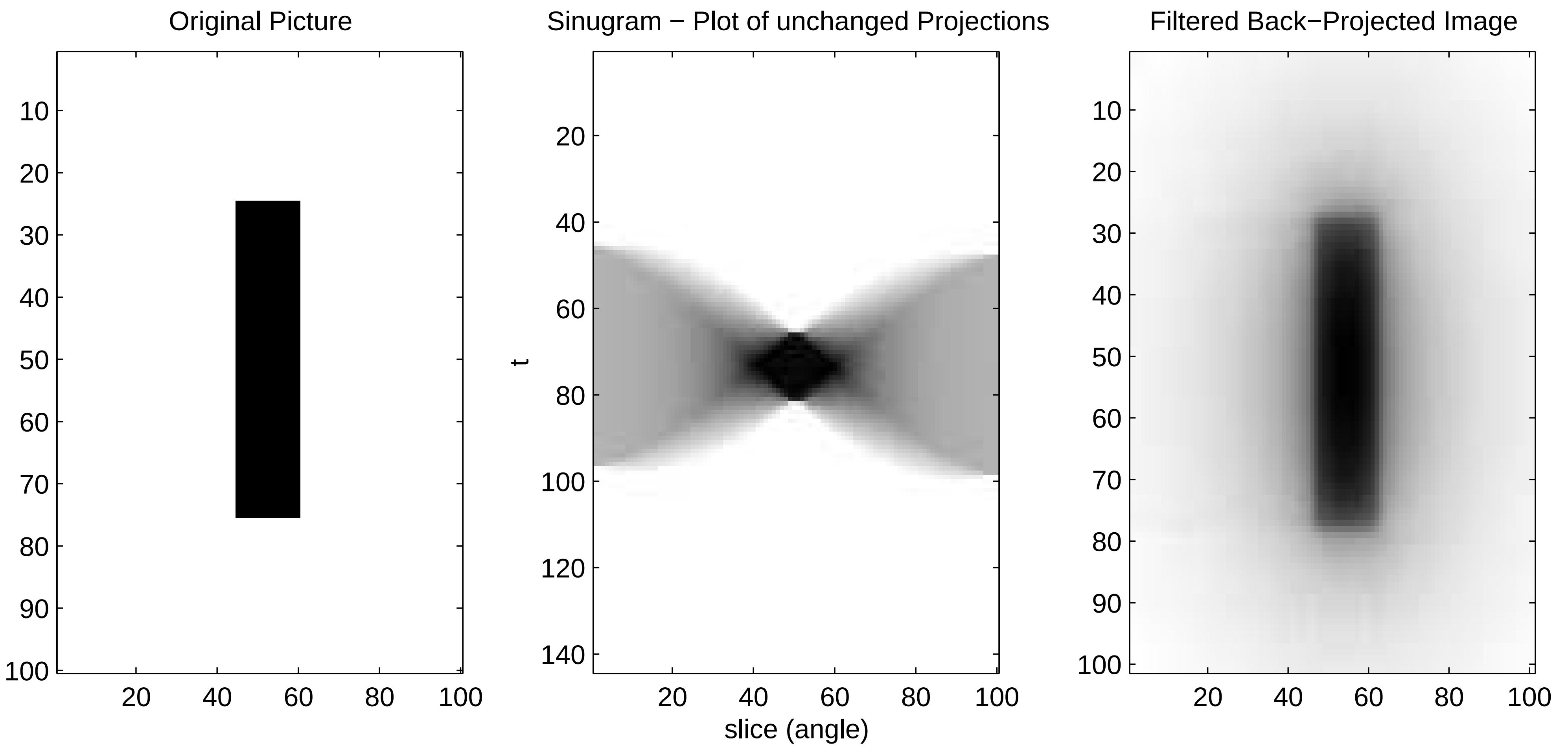
50 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

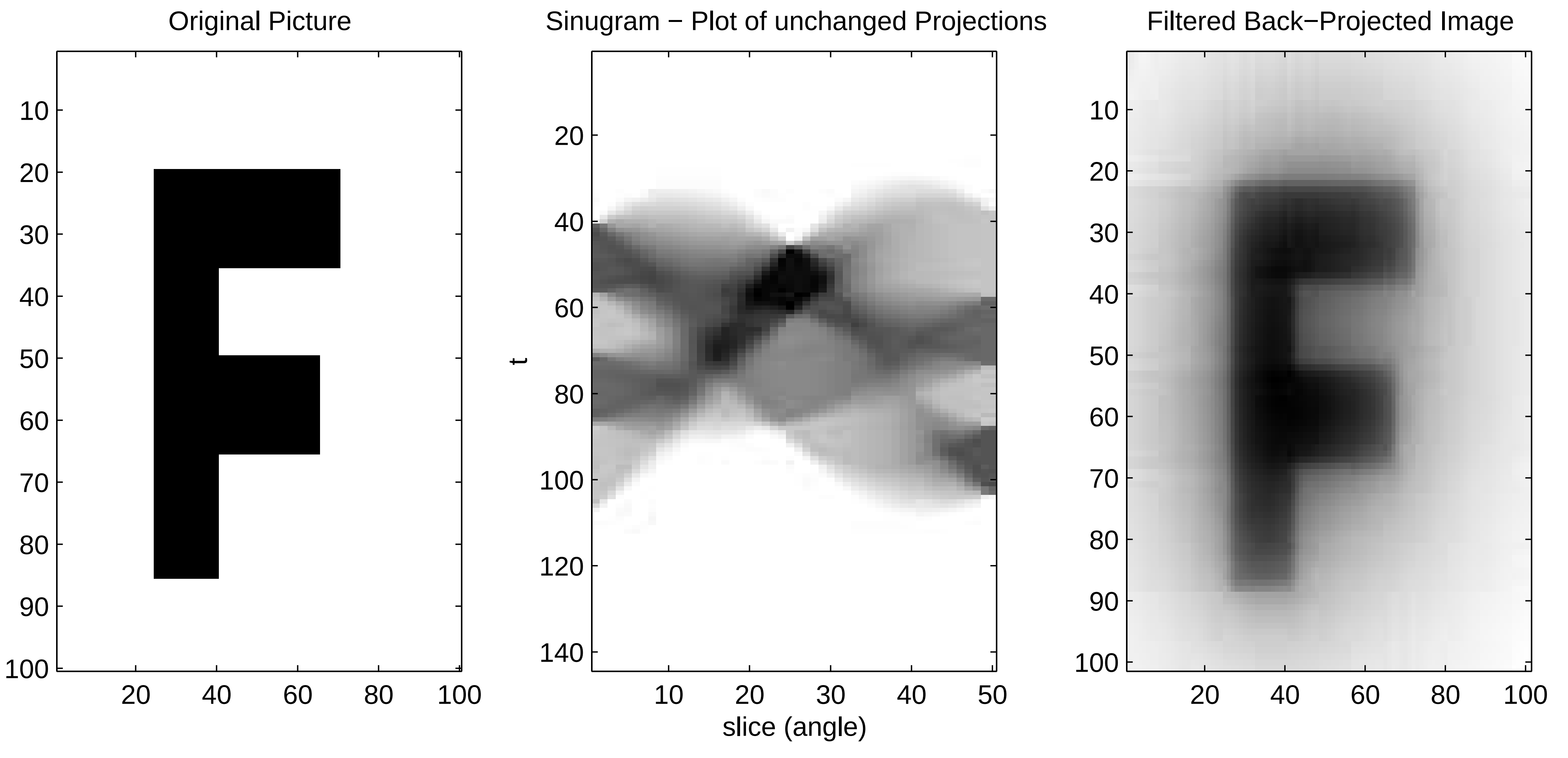
100 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

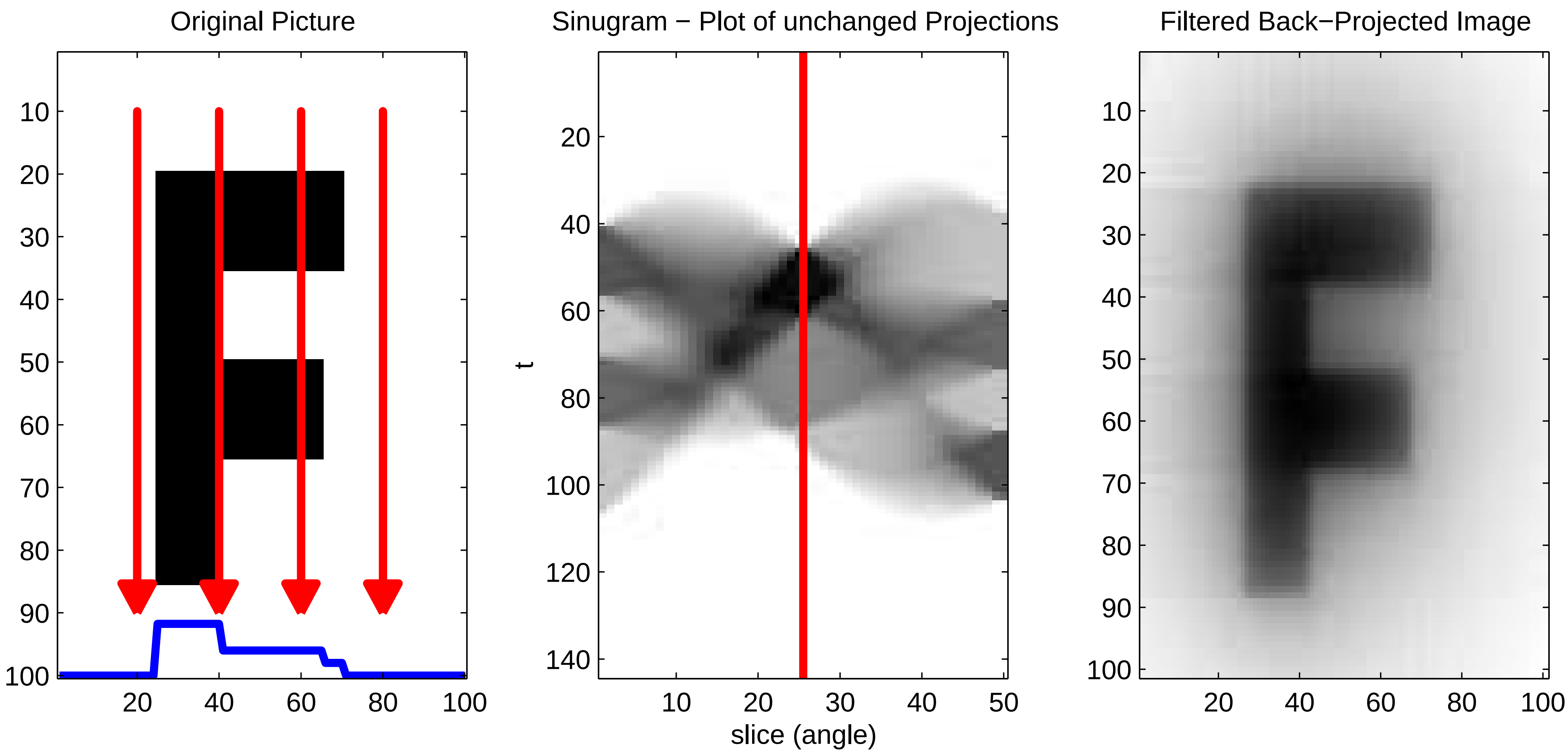
50 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm

Examples

50 projections



http://www.owlnet.rice.edu/~elec301/Projects00/tomography/code.htm