Image: 2-D function $x(i,j)$ where $0 \le i,j \le M-1$ (for an $M \times M$ image).

System: $\overset{\text{Continuous space (t,s)}}{\tilde{x}(t,s)} \rightarrow \underbrace{\overline{|A/D|}}_{\text{SAMPLER}} \rightarrow \underbrace{\overline{|h(i,j)|}}_{\text{FILTER}} \rightarrow \underbrace{\overline{|D/A|}}_{\text{INTERP–OLATOR}} \rightarrow \tilde{y}(t,s).$

D/A: $\tilde{y}(t,s) = 4\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} y(i,j)\left(\frac{B}{S}\right)^2 \frac{\sin B(t-iT)}{B(t-iT)}\frac{\sin B(s-jT)}{B(s-jT)}$ (2D sinc).

Sample: $t = iT, \quad s = jT, \quad S = \frac{2\pi}{T}, \quad S > 2B$ where $2B$=image bandwidth.

Assume: $X(\omega_1,\omega_2)) = \mathcal{FF}\{x(t,s)\} = 0$ for $|\omega_1| > B$ and $|\omega_2| > B$.

LTI: If $x_n(i,j) \rightarrow \overline{|LTI|} \rightarrow y_n(i,j)$, then $\sum c_n x_n(i,j) \rightarrow \overline{|LTI|} \rightarrow \sum c_n y_n(i,j)$.

h(i,j): If $x(i,j) \rightarrow \overline{|h(i,j)|} \rightarrow y(i,j)$, then $y(i,j) = \sum\sum h(i-m,j-n)x(m,n)$.
$h(i,j)$=impulse response=point-spread function.

Stable: BIBO stable $\Leftrightarrow \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}|h(i,j)| < \infty$.

2-D Z: $\mathcal{Z}\{h(i,j)\} = \sum\sum h(i,j)z_1^{-i}z_2^{-j} = H(z_1,z_2)$=transfer function.
$y(i,j) = h(i,j) * *x(i,j) \rightarrow Y(z_1,z_2) = H(z_1,z_2)X(z_1,z_2)$.

DSFT: $H(e^{j\omega_1},e^{j\omega_2}) = \sum\sum h(m,n)e^{-j(\omega_1 m+\omega_2 n)} = H(z_1,z_2)|_{z_1=e^{j\omega_1},z_2=e^{j\omega_2}}$.

Inverse: $h(m,n) = \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} H(e^{j\omega_1},e^{j\omega_2})e^{j(\omega_1 m+\omega_2 n)}d\omega_1\, d\omega_2$.

Note: $H(e^{j\omega_1},e^{j\omega_2})$ is periodic in ω_1 and ω_2 with periods 2π.

Odd+ Even $x_e(i,j) = \frac{1}{2}(x(i,j)+x(-i,-j)) = DSFT^{-1}\{RE[X(e^{j\omega_1},e^{j\omega_2})]\}$.
$x_o(i,j) = \frac{1}{2}(x(i,j)-x(-i,-j)) = DSFT^{-1}\{jIM[X(e^{j\omega_1},e^{j\omega_2})]\}$.
Any $x(i,j) = x_e(i,j) + x_o(i,j)$. DSFT[real and even] is real and even.

Freq. $\cos(\omega_1 i+\omega_2 j) \rightarrow \overline{|h(i,j)|} \rightarrow |H(e^{j\omega_1},e^{j\omega_2})|\cos(\omega_1 i+\omega_2 j+ARG[H(e^{j\omega_1},e^{j\omega_2})])$.

Resp. $|H(e^{j\omega_1},e^{j\omega_2})|$=Gain; $ARG[H(e^{j\omega_1},e^{j\omega_2})]$=Phase (shift).

2-D DFT: $X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1,n_2)e^{-j2\pi(\frac{n_1k_1}{N_1}+\frac{n_2k_2}{N_2})}, \quad \begin{matrix}0\le k_1\le N_1-1\\ 0\le k_2\le N_2-1\end{matrix}$.

Inverse: $x(n_1,n_2) = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X_{k_1,k_2}e^{j2\pi(\frac{n_1k_1}{N_1}+\frac{n_2k_2}{N_2})}, \quad \begin{matrix}0\le n_1\le N_1-1\\ 0\le n_2\le N_2-1\end{matrix}$.

Cyclic: $y(i,j) = h(i,j)\textcircled{c}\textcircled{c}x(i,j) \Leftrightarrow Y_{k_1,k_2} = H_{k_1,k_2}X_{k_1,k_2}$.

0-pad: 2-D cyclic convolution can be 0-padded to 2-D linear convolution.

2-D FFT AND: $N_1[N_2$-point 1-D FFTs] to compute $\hat{x}(n_1,k_2)$ from $x(n_1,n_2)$.
$N_2[N_1$-point 1-D FFTs] to compute X_{k_1,k_2} from $\hat{x}(n_1,k_2)$.
$\rightarrow N_1\frac{N_2}{2}\log N_2 + N_2\frac{N_1}{2}\log N_1 = \frac{N_1N_2}{2}\log(N_1N_2)$ mults. (like 1-D).

Example: $h(i,j) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow H(e^{j\omega_1}, e^{j\omega_2}) = \begin{cases} \left(\frac{\sin(1.5\omega_1)}{\sin(0.5\omega_1)}\right) \cdot \left(\frac{\sin(1.5\omega_2)}{\sin(0.5\omega_2)}\right) = \\ (1+2\cos(\omega_1))(1+2\cos(\omega_2)) \end{cases}$

Note: **Separable**: $h(i,j) = h_1(i)h_2(j) \rightarrow H(e^{j\omega_1}, e^{j\omega_2}) = H_1(e^{j\omega_1})H_2(e^{j\omega_2})$.

Point: Separable system/image→2-D problem decouples to 1-D problems.

So? 2-D lowpass: $h(i,j) = h_L(i)h_L(j)$ where $h_L(i)$ is 1-D lowpass filter.

Radial: $H(e^{j\omega_1}, e^{j\omega_2}) = \begin{cases} 1 & \text{for } \sqrt{\omega_1^2+\omega_2^2} < B \\ 0 & \text{for } \sqrt{\omega_1^2+\omega_2^2} > B \\ & 0 \le |\omega_1|, |\omega_2| \le \pi \end{cases}$

Note: $H(e^{j\omega_1}, e^{j\omega_2})$ is circularly symmetric "pill-box function."

Recall: $H(e^{j\omega_1}, e^{j\omega_2})$ is periodic in ω_1 and ω_2 with periods 2π.

Then: $h(i,j) = \frac{B}{2\pi}\frac{J_1(BR)}{R}$ where $R = \sqrt{i^2+j^2}$. (Derivation: Lim p.57).

Where: $J_1(\cdot)$=Bessel function of 1^{st} kind of order 1. $\frac{J_1(x)}{x} \sim$ sinc function.

Note: $H(e^{j\omega_1}, e^{j\omega_2}) = H(\sqrt{\omega_1^2+\omega_2^2})$ for $\sqrt{\omega_1^2+\omega_2^2} < \pi \rightarrow h(i,j) = h(\sqrt{i^2+j^2})$

If: $H(e^{j\omega_1}, e^{j\omega_2})$=constant in rest of the region $0 \le |\omega_1|, |\omega_2| \le \pi$.

But: $h(i,j) = h(\sqrt{i^2+j^2})$ does **NOT** $\rightarrow H(e^{j\omega_1}, e^{j\omega_2}) = H(\sqrt{\omega_1^2+\omega_2^2})$.

MEDIAN FILTERING

For: Impulsive="salt-and-pepper"=shot noise (arises from bit errors).

while: Preserving image *edges*. **NOTE:** Example of *nonlinear* filtering.

1-D: $y(n) = \text{MEDIAN}[x(n+2), x(n+1), x(n), x(n-1), x(n-2)]$.

Why? Preserves edges: Note this has no effect on step function $u(n)$.

2-D: Apply 1-D median filter to each row, then to each column.

EDGE DETECTION

For: Enhancing edges. **Why?** Image segmentation, image understanding.

1. $\nabla^2 x(i,j) = x(i+1,j) + x(i-1,j) + x(i,j+1) + x(i,j-1) - 4x(i,j)$.
2. Find *zero-crossings* of *Laplacian* $\nabla^2 x(i,j) \Leftrightarrow$ gradient extrema.
3. If $\sigma_x^2(i,j) = \sum_{i-2}^{i+2}\sum_{j-2}^{j+2}[x(m,n) - \bar{x}(m,n)]^2 > \eta$, declare $\exists$"edge."

OR: Use *gradients* for *directional* edge detection:

1. $\nabla_x(i,j) = [x(i+1,j) - x(i-1,j)] + [x(i+1,j+1) - x(i-1,j+1)]$
   $+[x(i+1,j-1) - x(i-1,j-1)] \approx \frac{\partial x}{\partial i}$ (central differences).
2. If $|\nabla_x(i,j)| > \eta$, declare $\exists$"edge" in vertical direction.
3. Use only local maxima of $|\nabla_x(i,j)|$: "edge thinning."

Given: Radon transform $p(t,\theta) = \int\int f(x,y)\delta(t - x\cos\theta - y\sin\theta)dx\,dy$.
Goal: Reconstruct *image* $f(x,y)$ from its *projections* $p(t,\theta)$.
Note: No longer using discrete pixel representation $x(i,j)$ of image.

Why? Basic *tomography* problem: Image reconstruction from projections.
Example: CAT (Computed Axial Tomography)-scan (x-ray tomography):
$p(t,\theta)$=absorption of probing x-ray along the line $t = x\cos\theta + y\sin\theta$.
Then: $f(x,y)$=local absorption coefficient at (x,y); this is what is imaged.

SOLUTION: FILTERED BACKPROJECTION (FBP)

FBP: $f(x,y) = \frac{1}{2\pi}\int\int \mathcal{H}\frac{d}{dt}p(t,\theta)\delta(t - x\cos\theta - y\sin\theta)dt\,d\theta$; ($\mathcal{H}$=Hilbert).

where: $\mathcal{H}\frac{d}{dt}p(t) = \mathcal{F}^{-1}\{|k|\mathcal{F}\{p(t)\}\}$ and $|k| = (jk)(-j\text{SGN}[k])$.

Why? Compute $\mathcal{F}_{t\to k}$ of Radon transform→the *projection-slice theorem:*
$P(k,\theta) = \int\int f(x,y)e^{-jkx\cos\theta}dx\,e^{-jky\sin\theta}dy = F(k\cos\theta, k\sin\theta)$.

Next: Inverse 2-D Fourier transform of $F(k_x,k_y)$ in *polar coordinates:*
$f(x,y) = \frac{1}{4\pi^2}\int\int F(k\cos\theta, k\sin\theta)e^{j(kx\cos\theta+ky\sin\theta)}k\,dk\,d\theta$
$= \frac{1}{2\pi}\int \mathcal{H}\frac{d}{dt}p(t = x\cos\theta + y\sin\theta)d\theta$. QED.

1. Rotate around object↔vary θ. Must interpolate between angles θ_i.
2. *Direct Fourier method:* Interpolate $F(k\cos\theta, k\sin\theta)$ on polar raster.
3. Algebraic Reconstruction Technique (ART): System of equations.
4. Synthetic Aperture Radar (SAR) ⇔ tomography: David Munson.

VITAL MATLAB COMMANDS

1. To load "mandrill.mat":>>**load mandrill.mat;A=xx(1:256,1:256);**
2. To load "mandrill.tif":>>**A=imread('mandrill.tif','tif');**
   Can handle the following formats: jpg, tif, gif, bmp, png, pcx
3. Display matrix A as image:>>**imagesc(A);colormap(gray);axis off**
   or: >>**imshow(A)** automatically gives you a grayscale image
4. **fft2(A,M,N);ifft2(A,M,N);conv2(H,X);filter2(A,B,X);freqz2(H)**
5. Vertical edge detection with threshold n:>>**edge(A,'sobel',n,'vertical')**
6. Laplacian-of-Gaussian edge detection:>>**edge(A,'log')**
7. 5×5 2-D median filtering:>>**medfilt2(A,5,5)** (default: 3×3)
8. Resize image:>>**Y=imresize(X,F)** Y is F times the size of X.
   Default method: nearest-neighbor interpolation.
9. Radon transform commands:>>**radon;iradon;phantom**
   Y=radon(X) default: each of 180 columns is projection at an angle.
   iradon uses the filtered backprojection method.
   phantom default: 256×256 modified Shepp-Logan head.

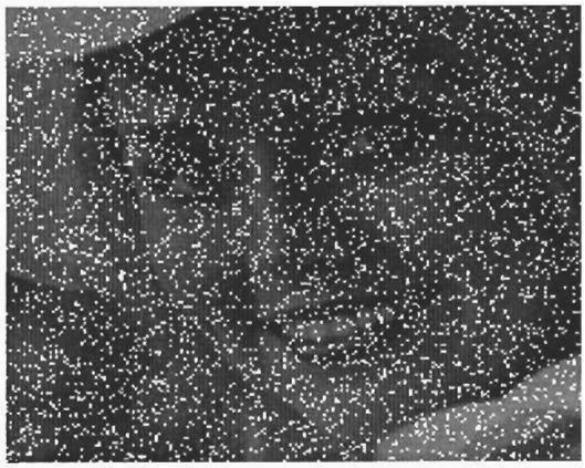
H=remez(20,[0,.15,.2,1],[1,1,0,0]);

edge(X,"sobel")

Y1=X+rand(201,201)/2;

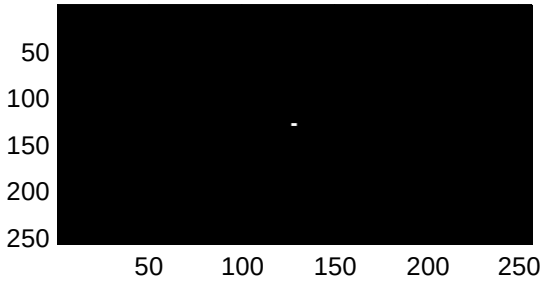
conv2(H^T·H,Y1)

Y2=X+floor(rand(201,201)+0.1);

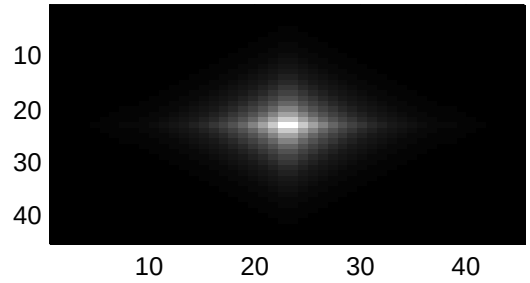


medfilt2(Y2)

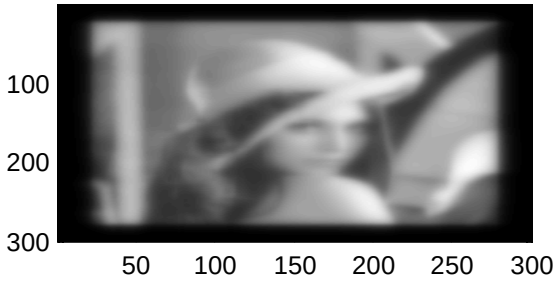
load lena.mat;X=xx;N=2*randn(300,300);2D IMPULSE



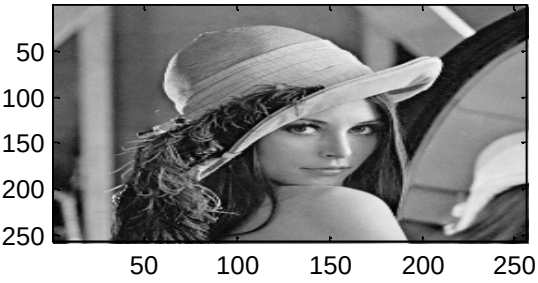
H=.8.^abs(-22:22);H=H*H; POINT-SPREAD



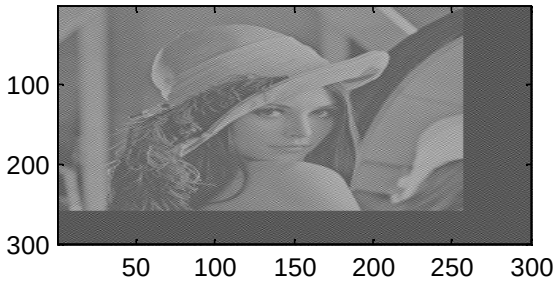
Y=conv2(X,H);Y1=Y+N;BLURRED IMAGE



real(ifft2(fft2(Y)./fft2(H,300,300)));RECONSTRUCTION



real(ifft2(fft2(Y1)./fft2(H,300,300)));RECONSTRUCTION



FX(76:225,76:225)=0;real(ifft2(FX));RECONSTRUCTION