MATLAB/Image Processing Toolbox

MATLAB's edge detection routines

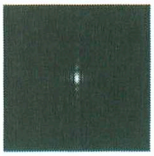
```
% [g,t] = edge (f, 'method', parameters)
%  f is input image, g is output image
% t is an optional threshold for the output image
% 'method' can be sobel, prewitt, roberts, laplacian of a gaussian,
% zero crossings, or Canny

>> f=imread('fig10.10(a).jpg');        %load in building figure
>> [g_sobel_default,0.074]=edge(f,'sobel');  % figure 10.7(a)
>> [g_log_default, 0.0025]=edge(f,' log');  % figure 10.7(c)
% log is short for laplacian of a Gaussian
>> [g_canny_default, [0.019,0.047]]=edge(f,'canny');  % figure 10.7(e)

% hand optimized functions
>> g_sobel_best=edge(f,'sobel', 0.05);  % figure 10.7(b)
%0.05 is a threshold for the output
>> g_log_best=edge(f,' log',0.003, 2.25);  % figure 10.7(d)
%0.003 is the output threshold and 2.25 is the standard deviation of the Gaussian
>> g_canny_best=edge(f,'canny', [0.04,0.10],1.5);  % figure 10.7(f)
%0.04 and 0.10 are the output thresholds and 1.5 is the standard deviation of the Gaussian
```

SEE GWE, Section 10.1.3 Edge Detection Using Function **edge**

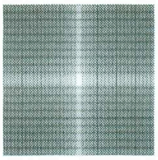
MATLAB/Image Processing Toolbox

MATLAB Fourier transforms

```
>> f=imread('Figure_Rectangle.jpg');  % load in spatial rectangle
>> F=fft2(f);                          % do 2D FFT
>> S=abs(F);                           % determine magnitude for disp
>> imshow(S, [ ])                      % shows in four corners of displa
>> Fc=fftshift(F);                     % shift FFT to center
>> imshow(abs(Fc), [ ]);               % show magnitude of FFT in cen
```



```
% much tougher to do display transform
>> g=im2unit8(mat2gray(log(1+double(f))));
>> imshow(g)
% double converts the image to double precision floating point
% mat2gray brings the values to the range [0,1]
% im2unit8 brings the values back to the range [0,255]

% general log transform
>> g=im2unit8(mat2gray(c*log(1+double(f))));
```

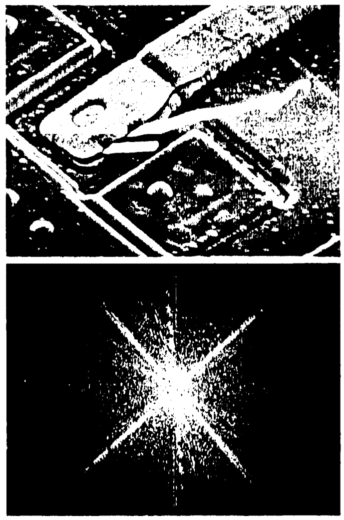


SEE GWE, Section 4.2 Computing and Visualizing the 2-D DFT in MATLAB
GWE, Section 3.2.2 Logarithmic and Contrast Stretching Transformations

Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain



a
b

FIGURE 4.4 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

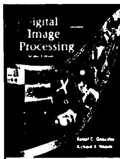
due to white oxide extrusions

value $F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$ which is the average image intensity

Note that $|F(u,v)| = |F(-u,-v)|$

In this figure the strong axes of $|F(u,v)|$ correspond to the approximate $\pm 45^\circ$ edges in the SEM image.

The angle of the white oxide extrusions wrt horizontal roughly corresponds to the short white off-center line in the transform.



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain

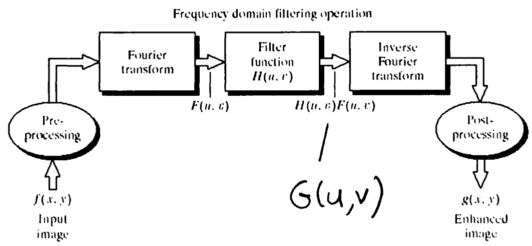


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Preprocessing examples
- $(-1)^{x+y}$ multiplication
- cropping image to even dimension
- gray level scaling
- conversion to floating point

H is known as a zero-phase filter. In general, filters for imaging are real and do not change the phase of $F(u,v)$

Notch filter

$$H(u,v) = \begin{cases} 0 & \text{if } (u,v) = \left(\frac{M}{2}, \frac{N}{2}\right) \\ 1 & \text{otherwise} \end{cases}$$

Such a filter is called a notch filter because it removes only $F(0,0)$ which is the average value of the image.

Basic filtering in the frequency domain

1. Multiply the input image by $(-1)^{x+y}$

$$\mathfrak{F}[f(x,y)(-1)^{x+y}] = F\left(u-\frac{M}{2}, v-\frac{N}{2}\right)$$

2. Compute $F(u,v)$ for the $(-1)^{x+y} f(x,y)$ using the DFT

This centers the transform of the image.

3. Multiply $F(u,v)$ by a filter function $H(u,v)$

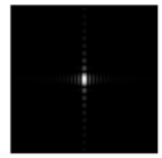
4. Compute inverse DFT of product

5. Take real part of inverse DFT

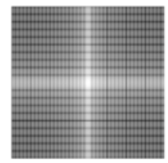
6. Multiply by $(-1)^{x+y}$ to get rid of (cancel out) initial multiplication

MATLAB/Image Processing Toolbox

MATLAB Fourier transforms



```
>> f=imread('Figure_Rectangle.jpg');  % load in spatial rectangle
>> F=fft2(f);                          % do 2D FFT
>> S=abs(F);                           % determine magnitude for display
>> imshow(S, [ ])                      % shows in four corners of display
>> Fc=fftshift(F);                     % shift FFT to center
>> imshow(abs(Fc), [ ]);               % show magnitude of FFT in center

% much tougher to do display transform
>> g=im2unit8(mat2gray(log(1+double(f))));
>> imshow(g)
% double converts the image to double precision floating point
% mat2gray brings the values to the range [0,1]
% im2unit8 brings the values back to the range [0,255]
```



```
% general log transform
>> g=im2unit8(mat2gray(c*log(1+double(f))));
```

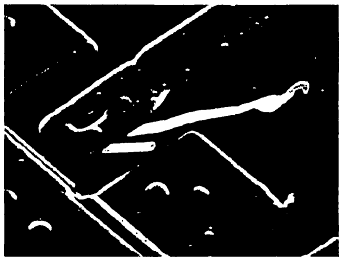
SEE GWE, Section 4.2 Computing and Visualizing the 2-D DFT in MATLAB
GWE, Section 3.2.2 Logarithmic and Contrast Stretching Transformations

Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain

FIGURE 4.6 Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the $F(0, 0)$ term in the Fourier transform.



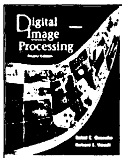
This is an example of a <u>displayed</u> image <u>after</u> F(0,0) was removed by a notch filter.

Since a display cannot show negative values the image has been shifted so that the most negative value corresponds to zero, i.e. black.

Other kinds of filters:

low pass – attenuates high frequencies while passing low frequencies which come from the large, smooth areas in the image. will show less gray scale variation.

high-pass – attenuates low frequencies while passing high frequencies which come from image detail such as edges and noise



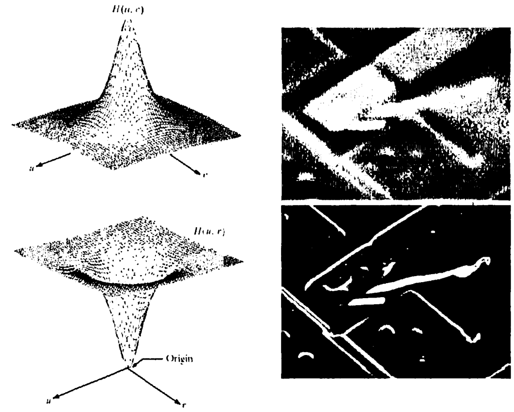
Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain

low-pass

high-pass

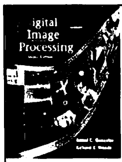


a b
c d

FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

(a) blurred image with high frequency detail missing due to low-pass filter.

(b) high-pass filter.
sharp image with prominent edge information
dark because negative information not being shown.

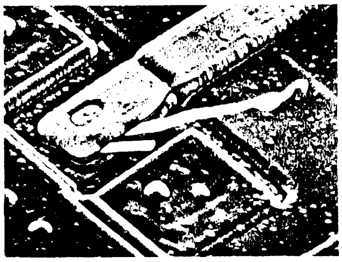


Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain

FIGURE 4.8 Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).

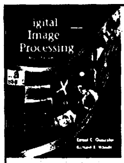


Result of adding a constant to the high-pass filtered image. Adding a constant shifts previously negative values from black into perceivable gray scale.

$$f(x,y) * h(x,y) \Longleftrightarrow F(u,v)H(u,v)$$

$$\delta(x,y) * h(x,y) \Longleftrightarrow \mathfrak{F}\left[\delta(x,y)\right] H(u,v)$$

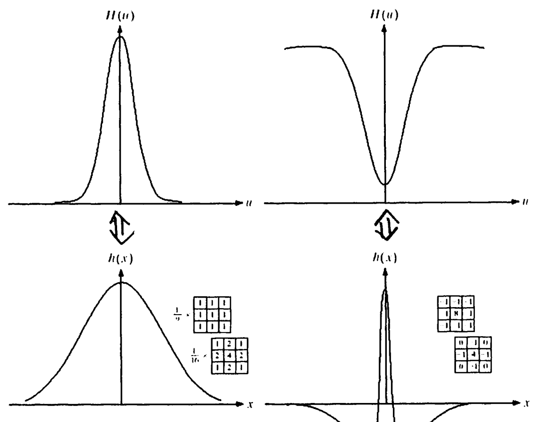
$$h(x,y) \Longleftrightarrow H(u,v)$$



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain



a b
c d

FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

what we have done is

$$H_{HP}(u,v) = 1 - H_{LP_2}(u,v)$$

approximated by $Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}}$

This is a difference of Gaussians

The Gaussian filter can be shown to be an optimal filter under certain conditions and has the property that both the forward and reverse transform of a Gaussian is a Gaussian.

$$h(x) = \sqrt{2\pi}\,\sigma\, A e^{-2\pi^2\sigma^2x^2}$$

$$H(u) = A e^{-\frac{u^2}{2\sigma^2}}$$

both real

construct using Pascal's triangle

```
        1
       1 1
      1 2 1
     1 3 3 1
    1 4 6 4 1
```

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

low-pass   weighted average filter from 3.6.1

Difference of Gaussians

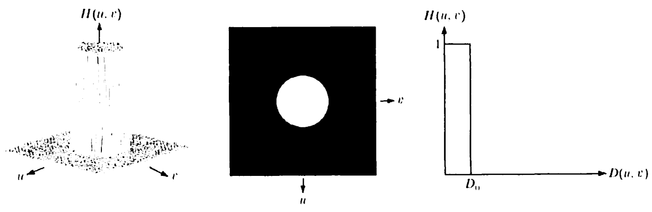
$$h(x) = \sqrt{2\pi}\,\sigma_1 A e^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\,\sigma_2 B e^{-2\pi^2\sigma_2^2x^2}$$

We usually design a filter in the frequency domain and implement it in the spatial domain.

Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Ideal Low Pass Filter
(ILPF)

Two-dimensional ideal low-pass filter

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) \ge D_0 \end{cases}$$

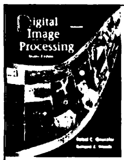
where $D(u,v) = \sqrt{(u-\frac{M}{2})^2 + (v-\frac{N}{2})^2}$ D_0 is the cutoff frequency and $D(u,v)$ is the radial distance from the center.

Pick cutoff frequencies based upon fraction of image power they remove.

$$P(u,v) = |F(u,v)|^2$$

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u,v)$$

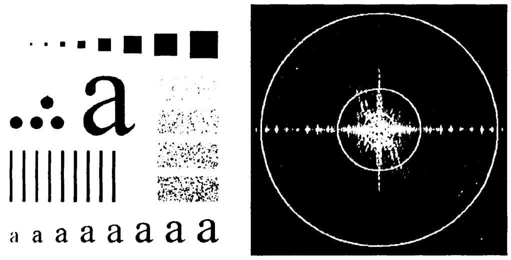
% of image power $\alpha = 100 \left[\sum_u \sum_v \frac{P(u,v)}{P_T} \right]$



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain



a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

80 230 255

There are five circles shown in the Fourier transform of this test image. These correspond to $\alpha = 92.0, 94.6, 96.4, 98$ & 99.5%

The corresponding u, v are 5, 15, 30, 80 & 230

The results of using each filter are shown in the next figure.

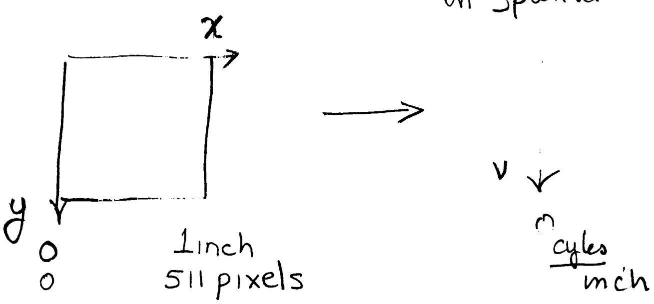
$$\Delta u = \frac{1}{M \Delta x}$$

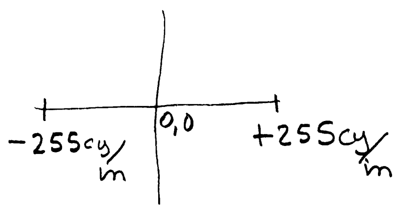
↑ spatial interval. Say $\frac{512 \text{ pixels}}{\text{inch}}$ Then $\frac{1}{512} \frac{\text{inches}}{\text{pixel}}$

= 0.001953 inch/pixel

$$\Delta u = \frac{1}{512 \times .001953} = 1 \text{ cycles/inch}$$

in spatial domain

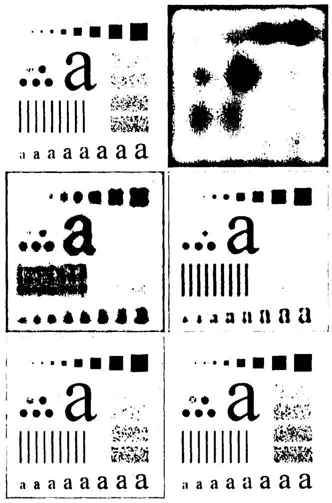




Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 4
Image Enhancement in the Frequency Domain



pretty useless except for blurring
all edge information contained in upper 8% of image power.

"ringing" shown in all images.

little edge information in upper 0.5% of power

a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Lowpass filtering

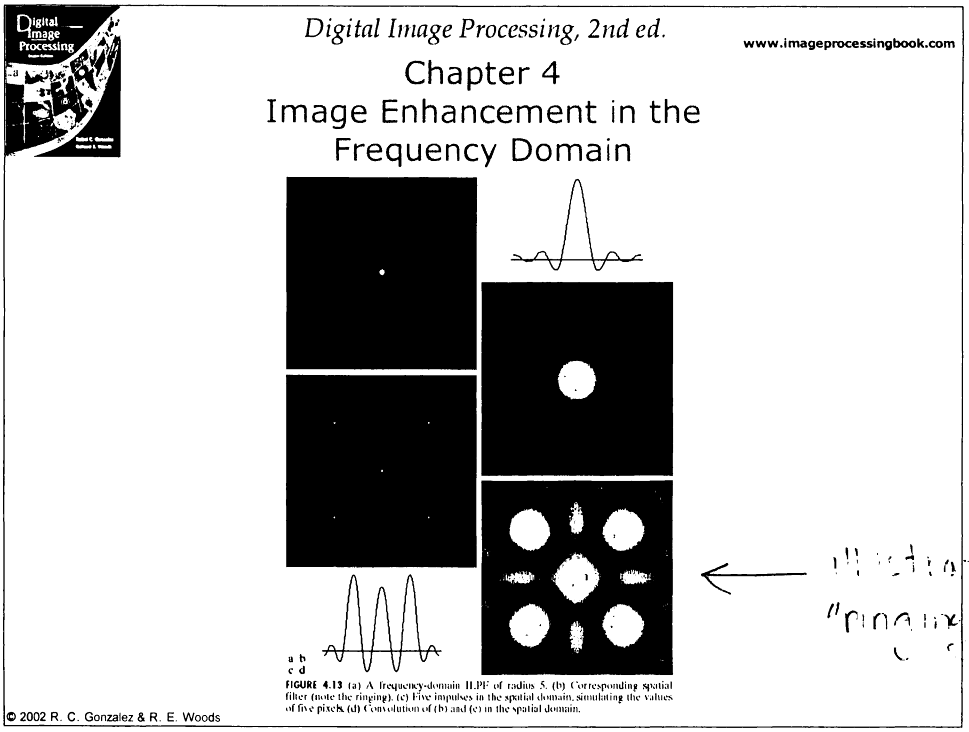


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

illustrates "ringing" due to ILPF (Ideal Low-Pass Filter)

(a) ideal low pass filter of radius 5 pixels

(b) corresponding spatial filter is computed

can do this in freq. domain $(-1)^{u+v} H(u,v)$ for de-centering (and get rid of complex shifts)

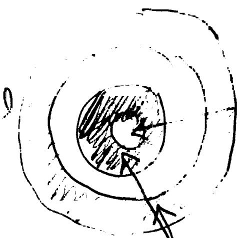
inverse DFT to get spatial filter

$(-1)^{x+y}\, R_e$ [inverse DFT]

re-centers

take real part since it should be real

As pixel radius of (a) increases the radius of the spatial rings decreases.



center is responsible for blurring

rings responsible for "ringing"

(c) 5 pixels in spatial domain "impulses"

(d) convolution of (b) with (c) showing "sifting" property

i.e., convolution of an impulse with a function "copies" the value of that function to the location of the impulse.