.

Filtering in the frequency domain

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it

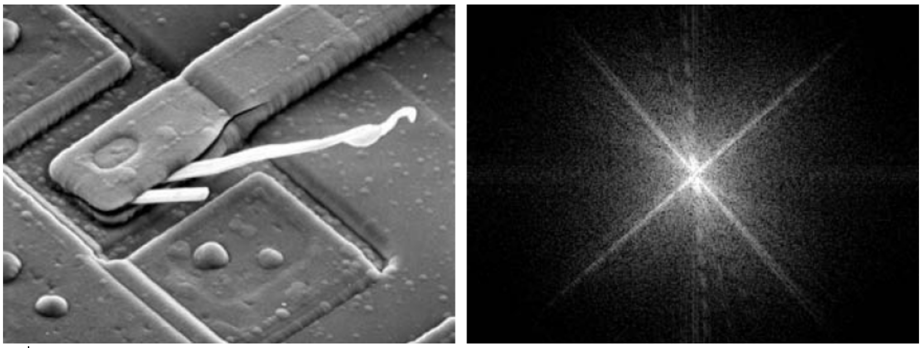
Methods for Image Processing

academic year 2018–2019

Fourier transform based analysis

- The FT of an image does not provide explicit information;
  - the FT components are the linear combination of all the elements of f.
- High frequencies are related to rapid variations, while low frequencies are related to slow variations (large scale).
- Some information on the structure of the scene (e.g., direction of the borders) can be deduced from the spectrum.

Fourier transform based analysis (2)



a | b

- In the spectrum (b) of the image (a) shows some large scale features are apparent: the $\pm 45°$ lines are the effect of the orientation of the main edges in the original image (a), while the nearly vertical component is produced by the bright oxide filament.

Filtering in the frequency domain

- Filtering in the frequency domain is operated by modifying the coefficients of the transformed image, and then transforming back the processed image.

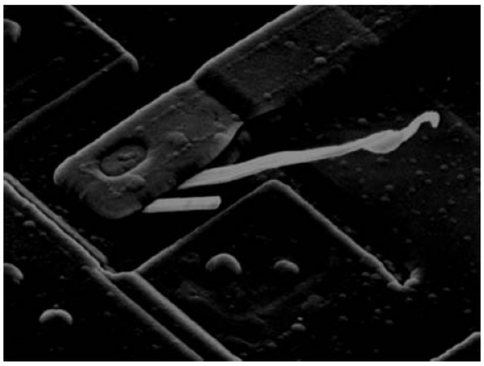
$$g(x, y) = \mathcal{F}^{-1}\{H(u, v)\, F(u, v)\}$$

- $H(u, v)$ is the *filter* function (or *filter transfer function*);
- $g(x, y)$ is the filtered image.
- F, H and g are arrays of the same size.
- If H is real and symmetric and f is real, g is real.
  - Imaginary components due to numeric errors can be ignored.

.

DC component

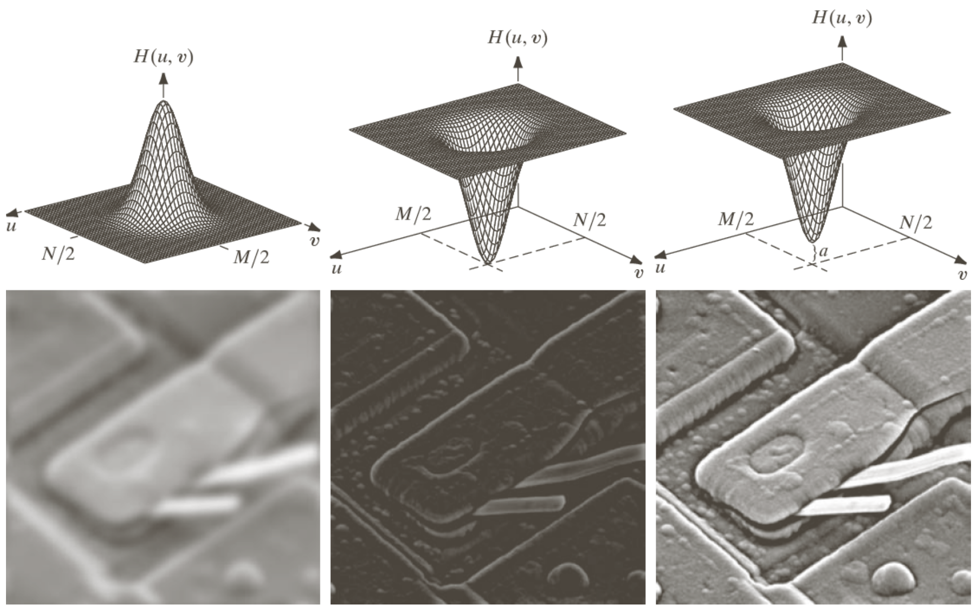
- The component of F corresponding to the origin of the frequencies, called *DC*, corresponds to the average intensity of f.
- Setting to zero only this component has the effect of shifting at zero the average of g.
  - If the gray levels are not rescaled, g will be darker.



Lowpass and highpass filters

- A *lowpass* filter attenuates the high frequencies and lets unaltered the low frequencies.
  - It will produce a defocused copy of the image.
- A *highpass* filter preserves the high frequencies and attenuates the low frequencies (the DC, in particular, should vanishes).
  - Improvement of the details, but the contrast is decreased.
  - Adding a constant, the DC are partially preserved.

Lowpass and highpass filters (2)



Wraparound and padding

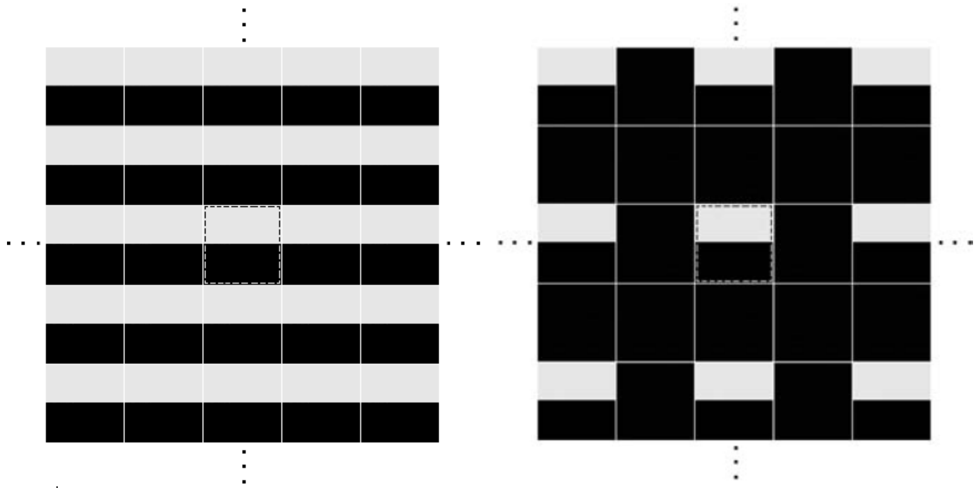
- The padding of the original image can avoid wraparound errors.
  - Both the image and the filter should be padded.
  - Padding have to be applied in the spatial domain.
- If the filter is specified in the frequency domain, it could be transformed in the spatial domain, padded, and transformed back in the frequency domain.
- However, *ringing* can arise and cause considerable errors.
- A better procedure consists in defining the filter in the frequency domain in the interval extended by the padding of the function.
  - The wraparound error are mitigated by the image padding and is preferable to ringing.

Wraparound and padding (2)

a | b | c

(b) and (c) are smoothed version of (a), respectively without and with padding.
Note the differences between (b) and (c) in the lateral and bottom sides.

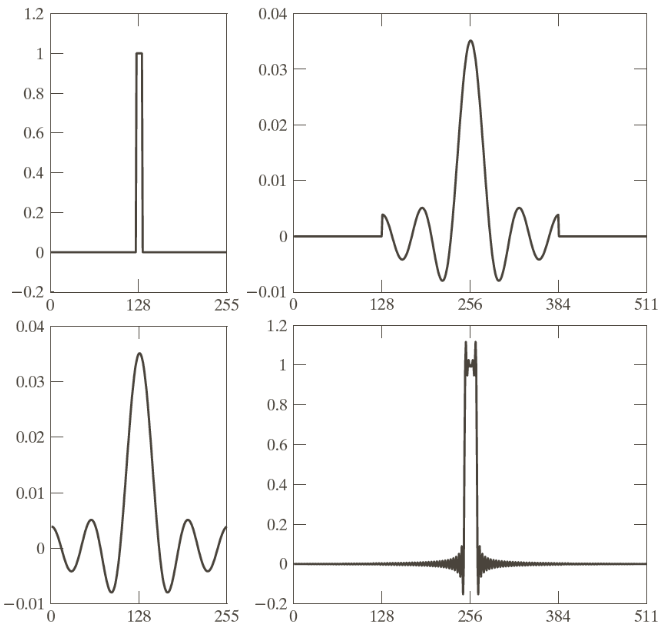
Wraparound and padding (3)



a | b

Inherent periodicity in images using DFT, (a) without and (b) with zero padding.

Wraparound and padding (4)



a	c
b	d

- Ideal filter (a) is transformed in the spatial domain (b), padded (c), and transformed back in the frequency domain (d).
- The effect of ringing in (d) is evident.

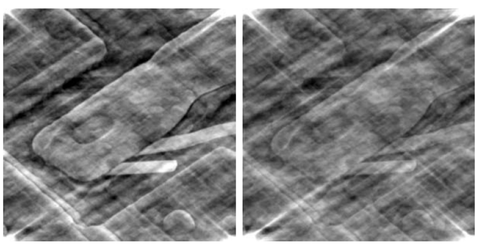
Zero phase shift filters

- Filtering can preserve the phase angle:

$$F(u, v) = R(u, v) + \iota I(u, v)$$

$$g(x, y) = \mathcal{F}^{-1}\{H(u, v)\, R(u, v) + \iota H(u, v)\, I(u, v)\}$$

- The ratio between real and imaginary parts does not change.
- Variations of the phase angle can have dramatic effect on the output of the filtering.
- Filters that do not change the phase angle are called *zero phase shift* filters.



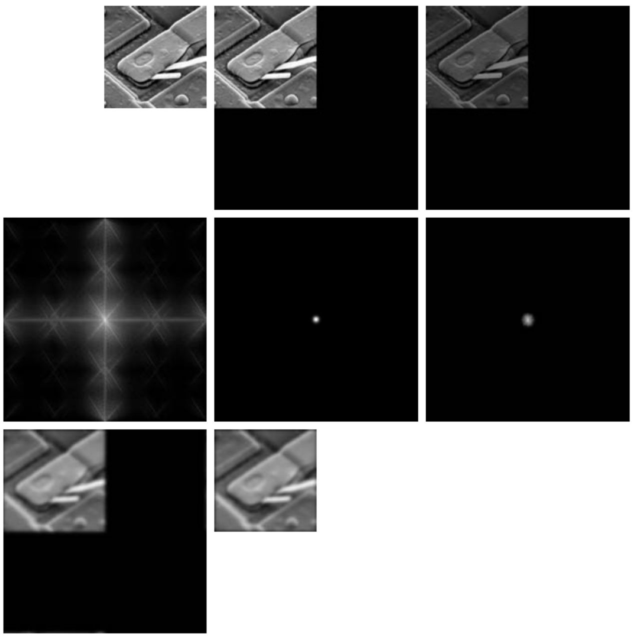
a | b

- Reconstruction using only the 50% (a) and the 25% (b) of the phase angle.

Summary

- Given $f(x, y)$ $[M \times N]$, P e Q are computed:
  - $P = 2M$ e $Q = 2N$
- Padding: $f \longrightarrow f_p$
- (optional) Multiplication of $f_p(x, y)$ by $(-1)^{x+y}$
- Computation of $F = \mathcal{F}\{f_p\}$
- Computation of the filter, $H(u, v)$ $[P \times Q]$ (optionally centered in (M, N))
- Computation of $G(u, v) = H(u, v)\,F(u, v)$
- $g_p(x, y) = \text{real}\left(\mathcal{F}^{-1}\{G(u, v)\}\right)$
- (optional) Multiplication of $g_p(x, y)$ by $(-1)^{x+y}$
- Cropping: the region $[M \times N]$ of g_p constitutes the filtered image, g.

Summary (2)



a	b	c
d	e	f
g	h	

(a) f
(b) f_p
(c) $f_p(-1)^{x+y}$
(d) F
(e) H
(f) G
(g) g_p
(h) g

.

Impulse response

- If $f(x, y) = \delta(x, y)$, $F(u, v) = 1$.
- Hence, $G(u, v) = H(u, v)\,F(u, v) = H(u, v)$, from which: $g(x, y) = h(x, y)$.
- $h(x, y)$ is called *impulse response* of $H(u, v)$.
- Since the all the quantities in the discrete implementation are finite, these filters are called Finite Impulse Response (FIR) filters.

Design of spatial filters

- Frequency domain knowledge can be used to guide the design of spatial filters.
- For example, the Gaussian filter can be considered:

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

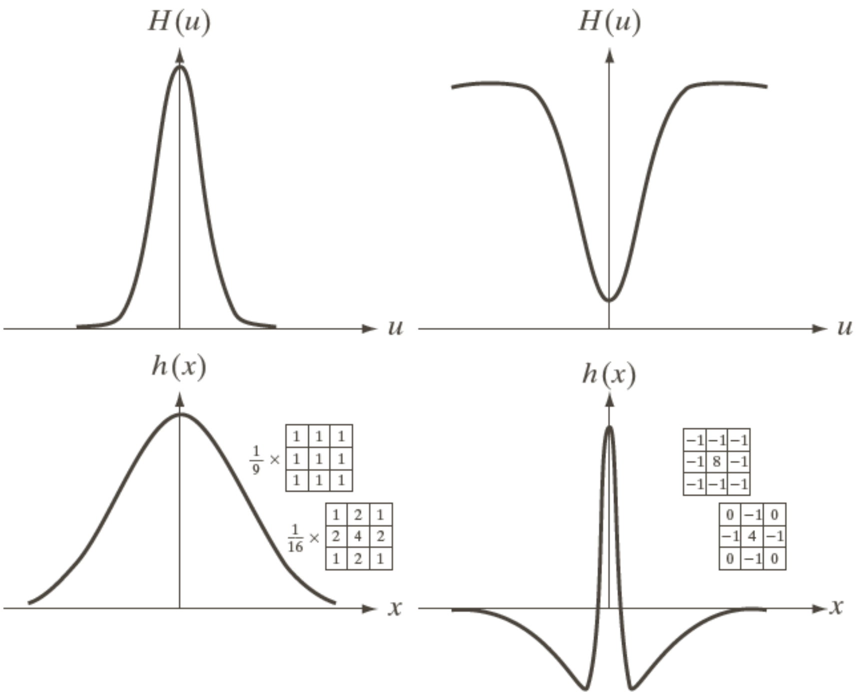
$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

- All the components are real and have a Gaussian behavior in both the domains.
- Their effects are intuitive.
  - Lowpass filtering with one Gaussian.
  - Highpass filtering with two Gaussians:

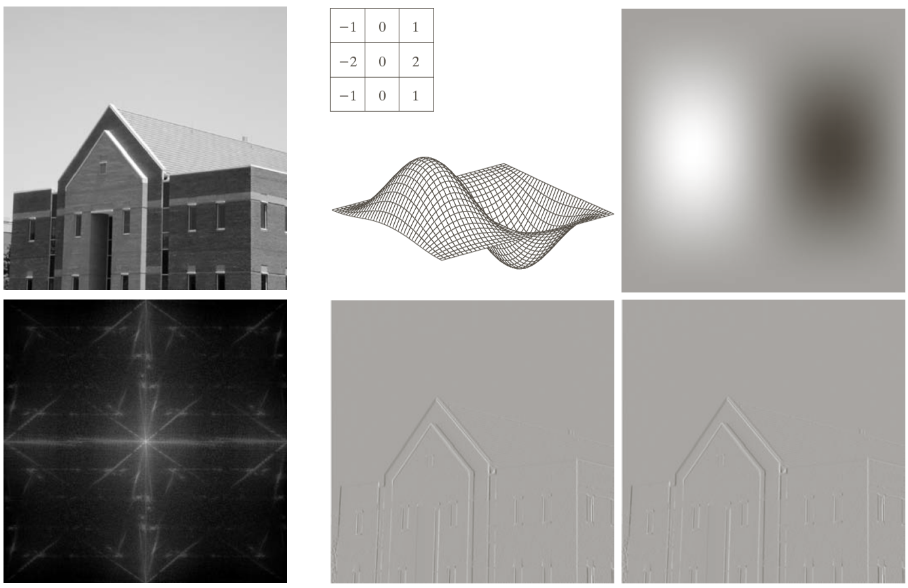
$$H(u) = Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}}$$

$$h(x) = \sqrt{2\pi}\sigma_1 Ae^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\sigma_2 Be^{-2\pi^2\sigma_2^2x^2}$$

Design of spatial filters (2)



Design of spatial filters (3)



Lowpass filters

- Ideal lowpass filter:

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

where H is the filter spectrum, D is the pixels distance function and D_0 is the *cut-off frequency*.

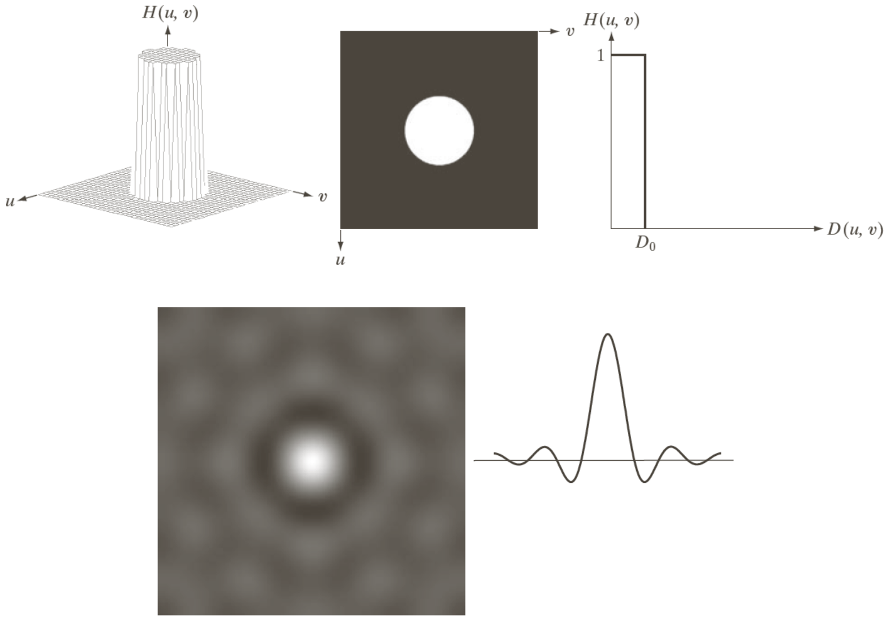
- Butterworth:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$

- Gaussian:

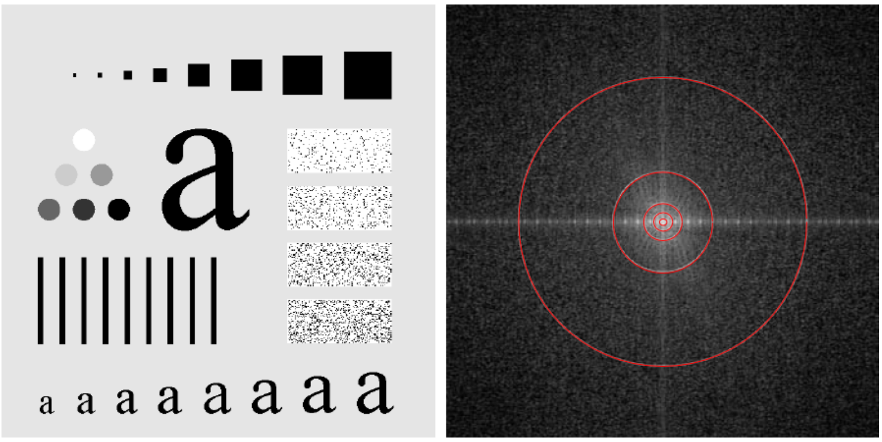
$$H(u, v) = e^{-\frac{D^2(u, v)}{2D_0^2}}$$

Ideal lowpass filter



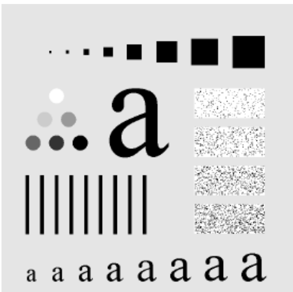
Ideal lowpass filter in frequency and spatial domain.

Test image



- Test image and its spectrum.
- The superimposed circles enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the spectrum power.
  - They correspond to the radii equal to 10, 30, 60, 160, 460.

Ideal lowpass filtering

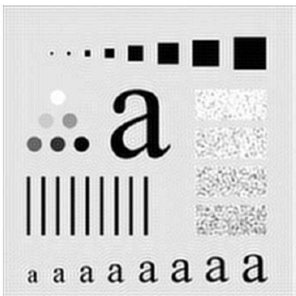


original

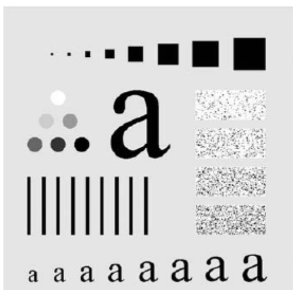
$D_0 = 10$

$D_0 = 30$

$D_0 = 60$

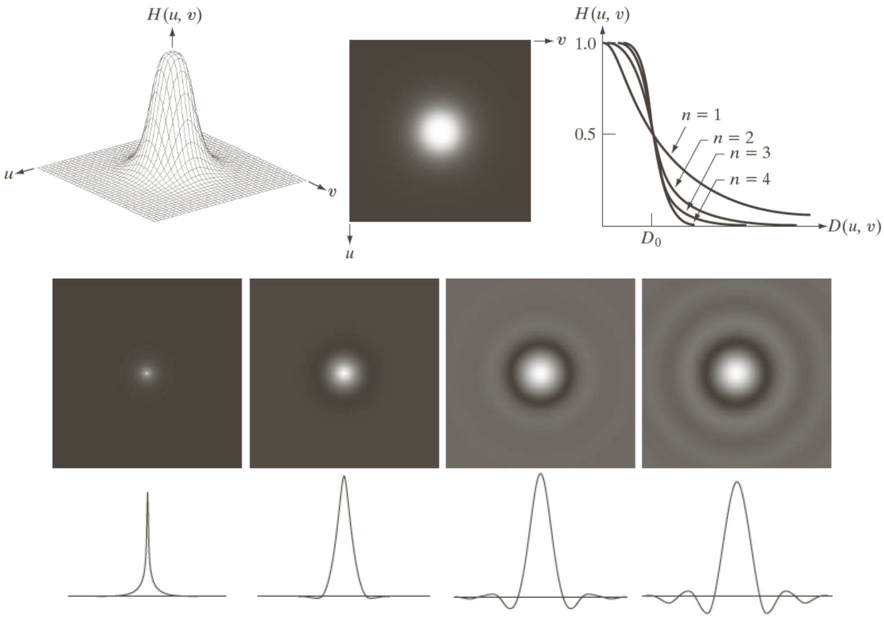


$D_0 = 160$



$D_0 = 460$

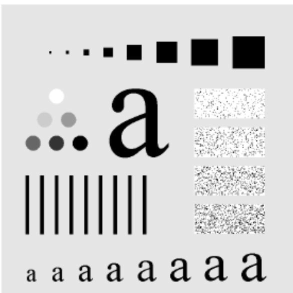
Butterworth lowpass filter



$n = 1$ $n = 2$ $n = 5$ $n = 20$

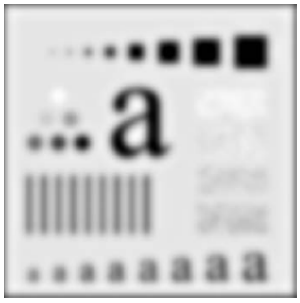
Butterworth lowpass filter in frequency and spatial domain.

Butterworth lowpass filtering



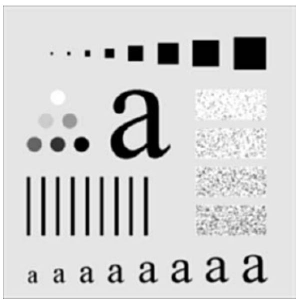
original

$D_0 = 10$

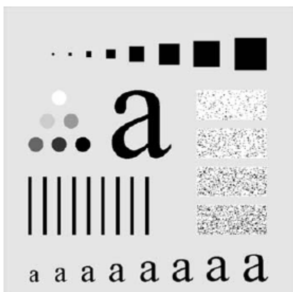


$D_0 = 30$

$D_0 = 60$

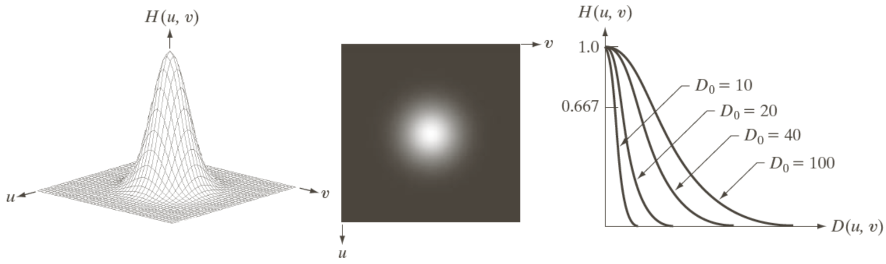


$D_0 = 160$



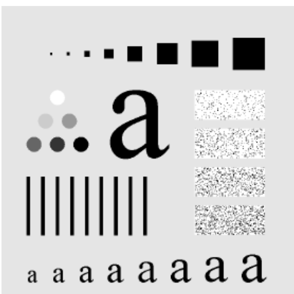
$D_0 = 460$

Gaussian lowpass filter



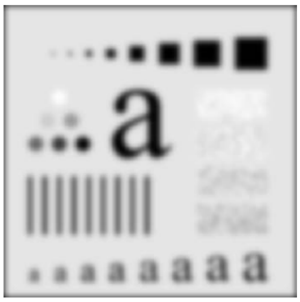
Gaussian lowpass filter in the frequency domain.

Gaussian lowpass filtering



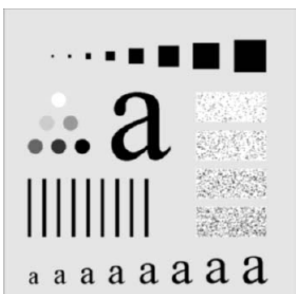
original

$D_0 = 10$

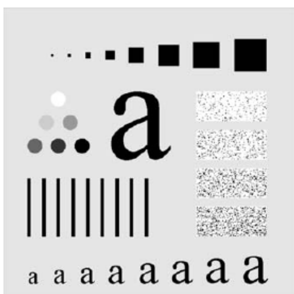


$D_0 = 30$

$D_0 = 60$

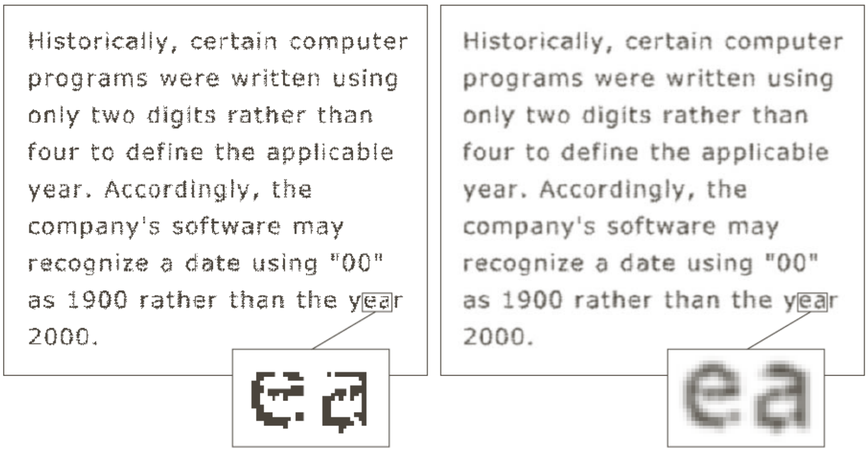


$D_0 = 160$



$D_0 = 460$

Lowpass filtering examples



a | b

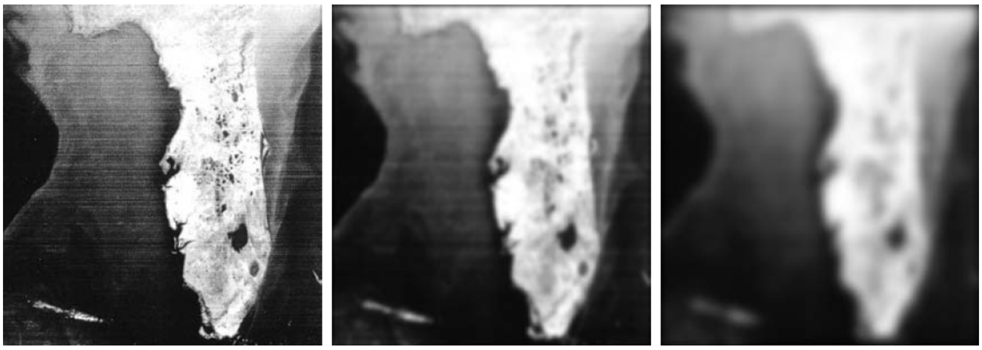
(a) Low resolution scanned text corrupted by noise.

(b) Lowpass filtering joins the parts of the broken characters.

Lowpass filtering examples (2)

- Aesthetic effects of the lowpass filtering.

Lowpass filtering examples (3)



- Lowpass filtering can eliminate the noise in the acquired image.
- As the cut-off frequency decreases, the smaller details are progressively lost.

Highpass filters

- Ideal highpass filter:

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

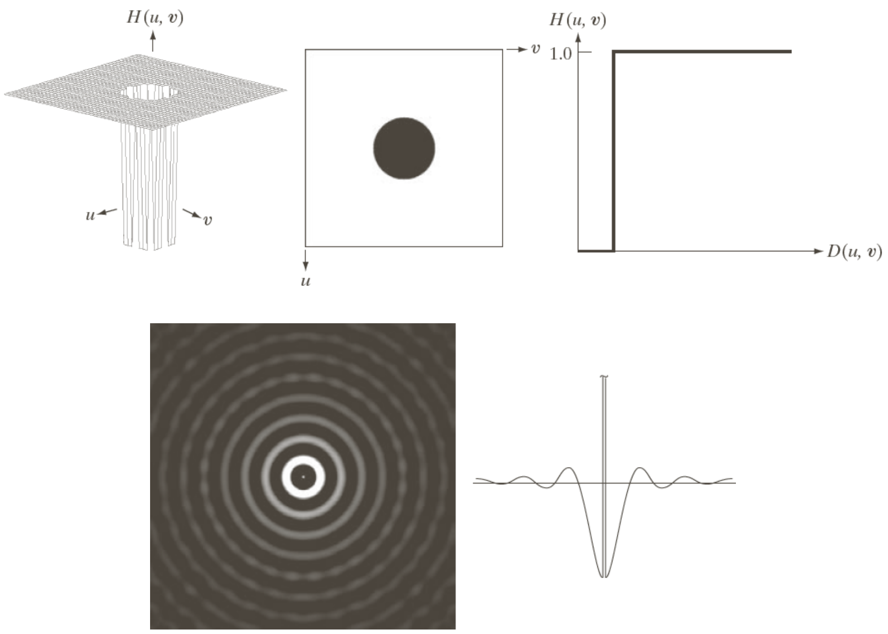
- Butterworth:

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

- Gaussian:

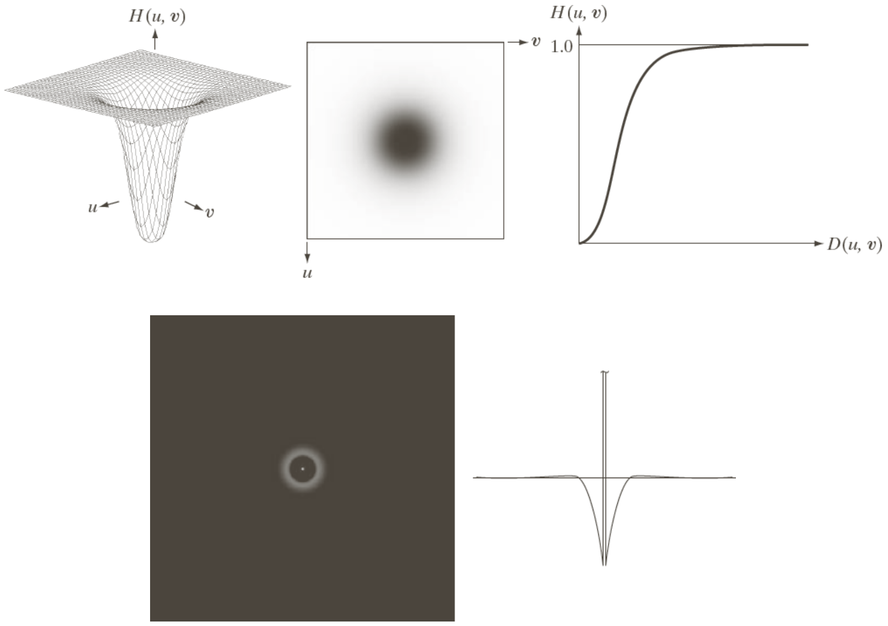
$$H(u, v) = 1 - e^{\frac{D^2(u, v)}{2D_0^2}}$$

Ideal highpass filter



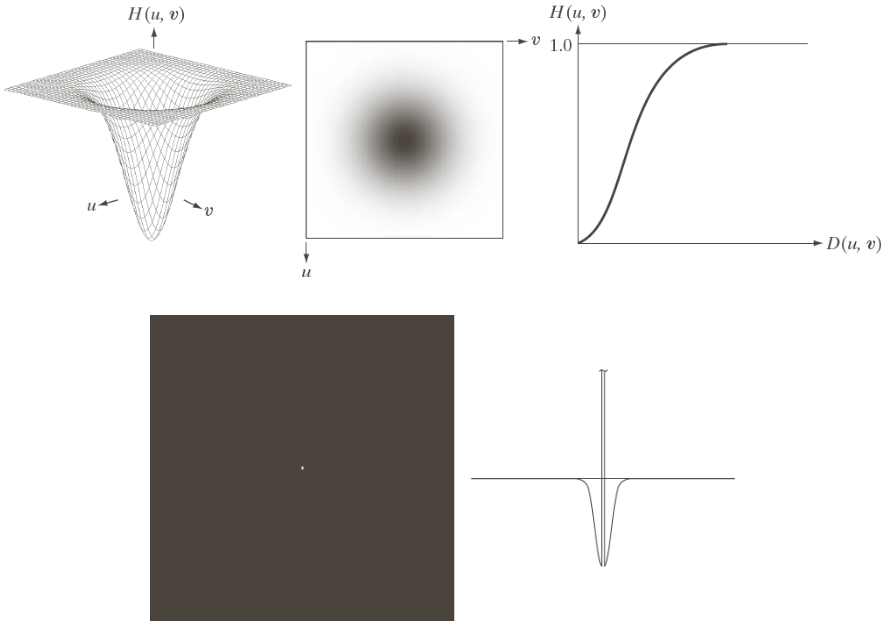
Ideal highpass filter in frequency and spatial domain.

Butterworth highpass filter



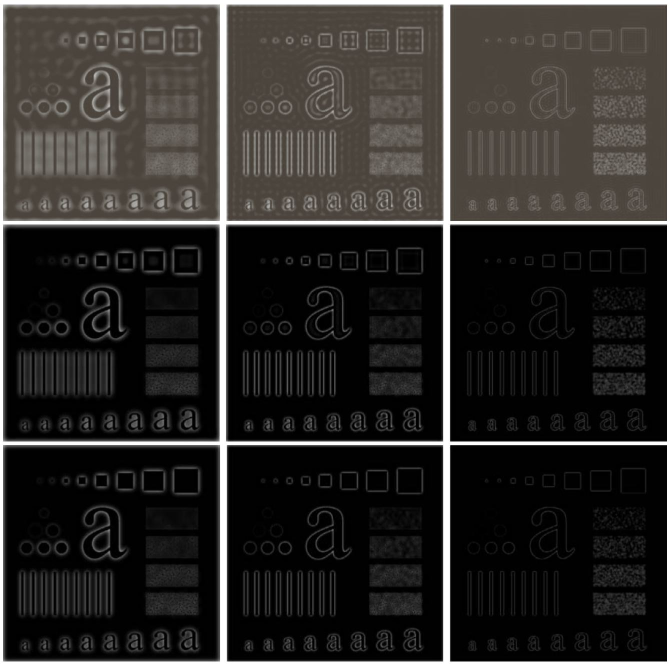
Butterworth highpass filter in frequency and spatial domain.

Gaussian highpass filter



Gaussian highpass filter in frequency and spatial domain.

Highpass filtering



	$D_0 = 30$	$D_0 = 60$	$D_0 = 160$
Ideal			
Butterworth			
Gaussian			

.

Highpass filtering example

a | b | c

(a) Thumb print image
(b) Result of highpass filtering
(c) After thresholding

Laplacian

- The Laplacian filter in the frequency domain is:

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

-

$$\nabla^2 f(x, y) = \mathcal{F}^{-1}\{H(u, v)\,F(u, v)\}$$

-

$$g(x, y) = f(x, y) + c\,\nabla^2 f(x, y)$$

Laplacian enhancement example

a | b | c

(a) Original blurry image
(b) Result of the Laplacian enhancement in the frequency domain
(c) Result of the Laplacian enhancement in the spatial domain

Unsharp masking

- The unsharp masking technique requires a mask g_{mask}:

$$g_{\text{mask}}(x, y) = f(x, y) + f_{\text{LP}}(x, y)$$

  where:

$$f_{\text{LP}}(x, y) = \mathcal{F}^{-1}\{H_{\text{LP}}(u, v)\, F(u, v)\}$$

- The filtered image results:

$$g(x, y) = f(x, y) + k\, g_{\text{mask}}(x, y)$$

- The process can be reframed as:

$$g(x, y) = \mathcal{F}^{-1}\{(1 + k\,(1 - H_{\text{LP}}(u, v)))\ F(u, v)\}$$

Unsharp masking (2)

- It can be expressed in terms of a highpass filter:

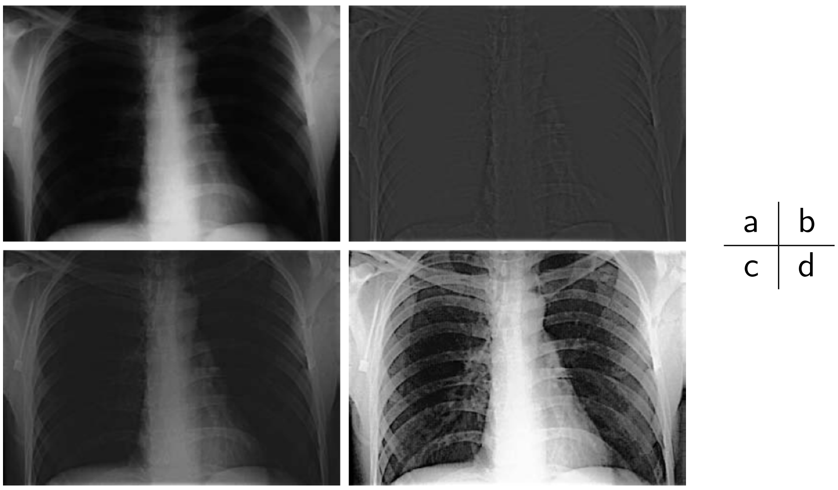
$$g(x, y) = \mathcal{F}^{-1}\{(1 + k\, H_{\mathrm{HP}}(u, v))\; F(u, v)\}$$

- Generalizing:

$$g(x, y) = \mathcal{F}^{-1}\{(k_1 + k_2\, H_{\mathrm{HP}}(u, v))\; F(u, v)\}$$

where:
  - $k_1 \geq 0$ is the offset from the origin,
  - $k_2 \geq 0$ is the high frequencies contribution.

Unsharp masking enhancement example



(a) Original X-ray image
(b) Result of Gaussian highpass filtering
(c) Result of unsharp masking using the same filter
(d) After histogram equalization on (c)

.

Homomorphic filtering*

- The intensity function of a scene, f, can be modeled as the composition of the illumination, i, and the reflectance, r:

$$f(x, y) = i(x, y)\, r(x, y)$$

- This relation cannot be exploited directly for the filtering in the frequency domain:

$$\mathcal{F}\{f(x, y)\} \neq \mathcal{F}\{i(x, y)\}\, \mathcal{F}\{r(x, y)\}$$

- A transformation able to separate the two components in the frequency domain has to be devised.

Homomorphic filtering* (2)

- The logarithm has the interesting property:

$$z(x, y) = \log f(x, y) = \log i(x, y) + \log r(x, y)$$

- Due to the linearity of the DFT:

$$Z(u, v) = \mathcal{F}\{z(x, y)\} = \mathcal{F}\{\log i(x, y)\} + \mathcal{F}\{\log r(x, y)\}$$

- The filter H can be applied to both the components:

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)F_i(x, y) + H(u, v)F_r(x, y)$$

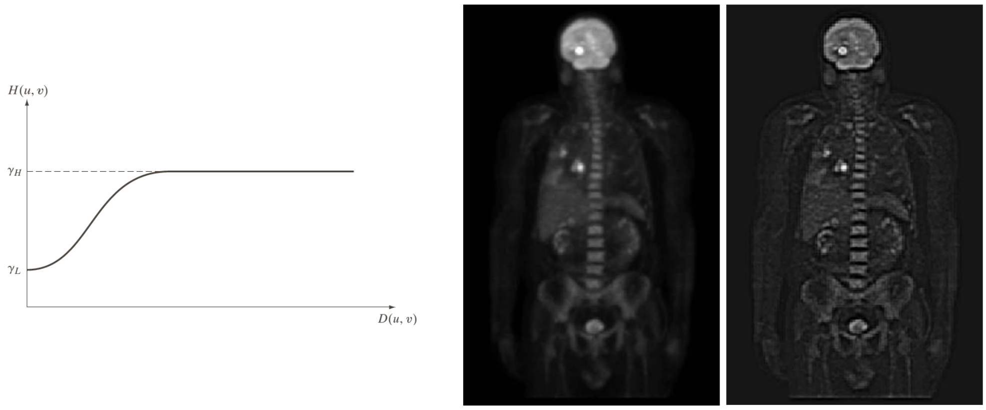
- In the spatial domain, the logarithmic transformation can be reversed:

$$g(x, y) = e^{\mathcal{F}^{-1}\{S(u, v)\}}$$

- The process can be operated using a filter such as:

$$H(u, v) = (\gamma_H - \gamma_L)\left(1 - \exp\left(-c\frac{D^2(u, v)}{D_0^2}\right)\right) + \gamma_L$$

Homomorphic filtering example*

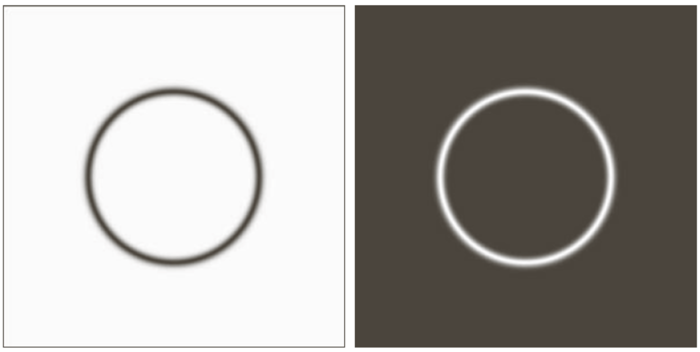


a | b | c

(a) Homomorphic filter
(b) Full body PET scan image
(c) Enhancement through homomorphic filtering

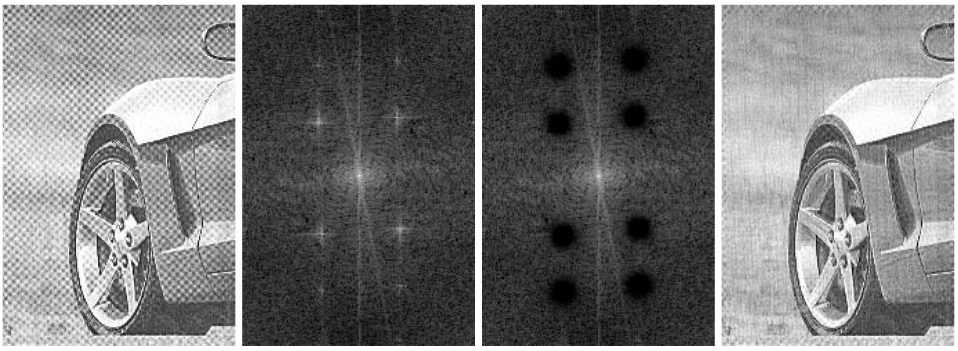
Bandreject and bandpass filters*

- Some applications operate on regions of the frequency domain:
  - selective filtering.
- Filters that operate on only some bands, they are called bandreject or bandpass filters.



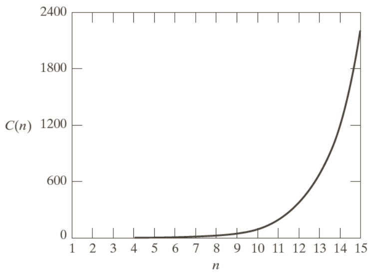
Notch filters*

- A notch filter operates on a limited neighborhood of some frequencies.
- Such regions must be symmetrical wrt the origin (zero phase shift).



DFT computation

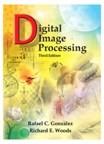
- The DFT is separable: monodimensional DFT of the rows, followed by monodimensional DFT of the columns.
- The IDFT is easily reframed in terms of DFT.
- The DFT complexity is $O((MN)^2)$.
- However, the DFT can be implemented through the *Fast Fourier Transform* (FFT) algorithm, which has a $O(MN \log_2(MN))$ complexity.
- Its implementation would require that M and N are power of 2, but it can be generalized.



- Computational advantage of FFT vs. direct implementation of DFT as a function of the number of samples.

.

Homeworks and suggested readings



DIP, Sections 4.7–4.9, 4.11

- pp. 255–293, 298–303