Filtering in the frequency domain

Stefano Ferrari

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

Methods for Image Processing

academic year 2015-2016





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.

<text><list-item><list-item>

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.



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.









Summary

- Given f(x, y) [M × 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 × Q] (optionally centered in (M, N))
- Computation of G(u, v) = H(u, v) F(u, v)
- $g_p(x, y) = \operatorname{real} \left(\mathcal{F}^{-1} \{ G(u, v) \} \right)$
- (optional) Multiplication of $g_p(x, y)$ by $(-1)^{x+y}$
- Cropping: the region [M × N] of g_p constitutes the filtered image, 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:

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

- 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:

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





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:

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





















Highpass filters • Ideal highpass filter: $H(u, v) = \begin{cases} 0, & D(u, v) \le D_0 \\ 1, & D(u, v) > D_0 \end{cases}$ • Butterworth: $H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$ • Gaussian: $H(u, v) = 1 - e^{\frac{D^2(u, v)}{2D_0^2}}$















Unsharp masking

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

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

where:

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

► The filtered image results:

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

► The process can be reframed as:

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



- (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:

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









Homeworks and suggested readings



DIP, Sections 4.7-4.9, 4.11

▶ pp. 255–293, 298–303