5. FREQUENCY ANALYSIS

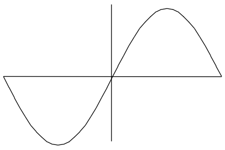
Image representations in frequency domain

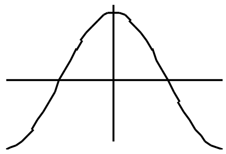
Any signal functions (including images, which are 2-dimensional signal functions) can be expressed as the integral (or sum) of simple periodic functions such as e.g. *sin* and *cos*. A set of basic (orthogonal[1]) periodic functions in terms of which the approximation is expressed is called the *basis functions* set. If the basis functions are B_0, B_1, B_2, …, a function f(x) can be approximated by the expression:

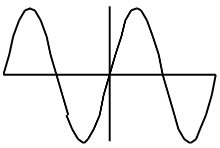
$$f(x) = \int_0^\infty c_i B_i \approx \sum_{i=0}^{\infty} c_i B_i = c_0 B_0 + c_1 B_1 + c_2 B_2 + \ldots$$

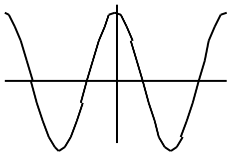
where c_i are appropriate coefficients.

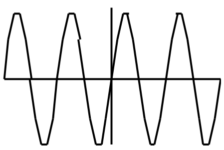
There can be many different sets of basis functions. In the most commonly used frequency transform, the *Fourier transform*, the basis functions are trigonometric functions *sin* and *cos* with different *frequencies*, *amplitudes* and *phases*.

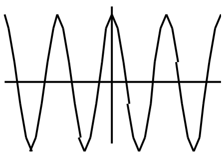
(a) *sin x*


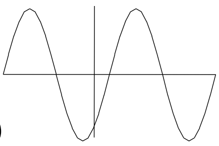
(b) *cos x*


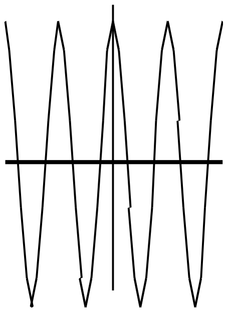
(c) *sin 2x*


(d) *cos 2x*


(e) *sin 4x*


(f) *cos 4x*


(g) *sin 2(x+π/4)*


(h) *2cos 4x*


Frequency can be thought of as the number of oscillations (changes) of brightness that occur in a unit of space; the more changes the higher the frequency. For example wave (e) has higher frequency than wave (c); and (c) than (a). Waves (c) and (d) have the same frequency.

Amplitude is a magnitude of change. For example, (f) and (h) have the same frequency, but the amplitude of (h) is higher.

[1] "Orthogonal" in math-speak means roughly that the functions are independent and they are the basis for defining other functions. For example the two axes of a 2-dimensional coordinate system, X and Y (i.e. vectors [1 0] and [0 1]) are independent and orthogonal.

Phase is a “shift” of a basis function from its “standard” position. For example (a) and (c) have the same phase (zero) although they have different frequency; (c) and (g) have the same frequency and amplitude, but different phase.

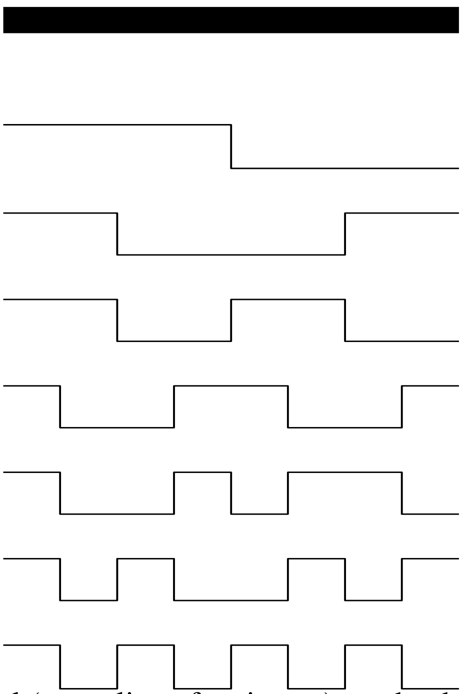
Frequency analysis

In order to analyse frequency characteristics of a signal, a *frequency transform* is applied. It is an algorithm which decomposes the data into a set of basis functions and thus it transforms the data from spatial domain to *frequency domain*.

Why do we want to decompose a signal into its frequency components? In the two previous Units we made intuitive observations that, for example, smooth and unchanging image areas represent low-frequency; edges and noise represent high frequency; smoothing removes high frequency components; during edge detection low frequency components are removed and high frequency components are retained. We learned how to use convolution to manipulate frequency characteristics of an image through appropriate manipulation of image values. One of the main advantages of representing images in frequency domain is that the frequency manipulations can be applied directly. This makes their use much more intuitive than indirect frequency manipulation via changes in pixel values.

Decomposition into frequency components

To explain the concept of decomposition of a signal into its frequency components let’s introduce a Walsh-Hadamard frequency transform. Unlike in Fourier transform (where basis functions are sinusoidal lines), the basis functions are rectangular waves, which can be easily interpreted as pixel values. Assuming for simplicity that the highest frequency is 8, the set of the Walsh-Hadamard basis functions is as follows:



Basis function	Plot	Values							
W(0,8)		1	1	1	1	1	1	1	1
W(1,8)		1	1	1	1	0	0	0	0
W(2,8)		1	1	0	0	0	0	1	1
W(3,8)		1	1	0	0	1	1	0	0
W(4,8)		1	0	0	1	1	0	0	1
W(5,8)		1	0	0	1	0	1	1	0
W(6,8)		1	0	1	0	0	1	0	1
W(7,8)		1	0	1	0	1	0	1	0

Any signal (e.g. a line of an image) can be decomposed into a linear combination of W(i,8) functions:

$$I(x) = \Sigma\, c_i\, W(i,8)$$

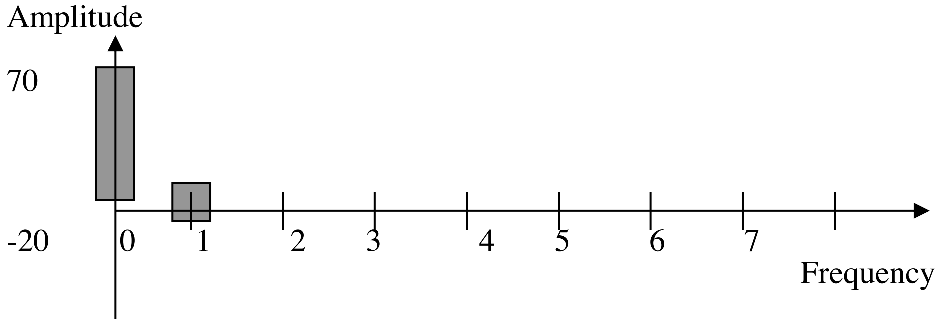
For example, a line in a plain image, where each pixel value is 50 can be represented as:

$I(x) = 50\cdot W(0,8) + 0\cdot W(1,8) + 0\cdot W(2,8) + 0\cdot W(3,8) + 0\cdot W(4,8) + 0\cdot W(5,8) + 0\cdot W(6,8) + 0\cdot W(7,8)$

A step edge from 50 to 70 (pixel values 50 50 50 50 70 70 70 70) can be represented as:
$I(x) = 70\cdot W(0,8)-20\cdot W(1,8) +0\cdot W(2,8) +0\cdot W(3,8) +0\cdot W(4,8) +0\cdot W(5,8) +0\cdot W(6,8) +0\cdot W(7,8)$

The decomposition does not have to be done manually! There are well established algorithms for doing so, for example Fast Fourier Transform, FFT (we are not going to study these algorithms in detail in this course).

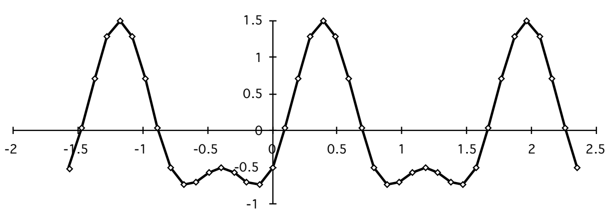
A plot of the resulting frequency domain representation of the data is also called a frequency transform. In this plot abscissas indicate frequency of basis functions and ordinates indicate amplitude. For example, in Walsh-Hadamard frequency transform can be represented as:

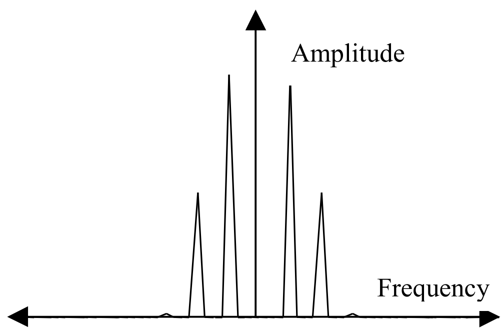


Example of decomposition, using Fourier transform, for a 1-dimensional signal in spatial domain:

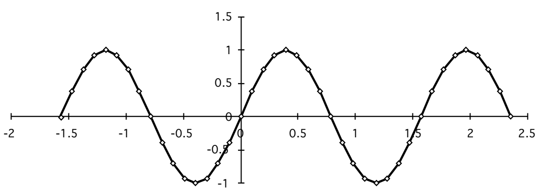
$$f(x) = \sin 4x - 0.5 \cos 8x$$

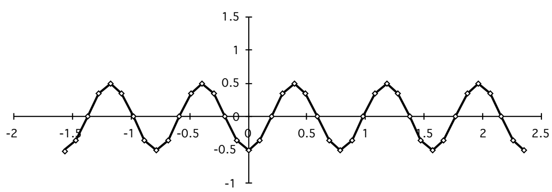
The plot of this function is shown in the graph on the left. Fourier Transform (FT) of the function has the plot shown on the right. In this particular form of FT, zero frequency is at the centre of the plot and frequency increases outwards, away from the centre. The plot is usually symmetric (either even or odd symmetry).





From the Fourier transform plot we can see that only two frequencies are present, these are:





$f(x) = \sin(4x)$ $f(x) = -0.5 \cos(8x)$

These two functions combined (i.e. added) together will result in the input function f(x).

The Fourier transform (for the discrete case) can be expressed by the formula:

$$F(u) = \sum_x f(x) e^{-j2\pi ux}$$

and the inverse transform by the formula:

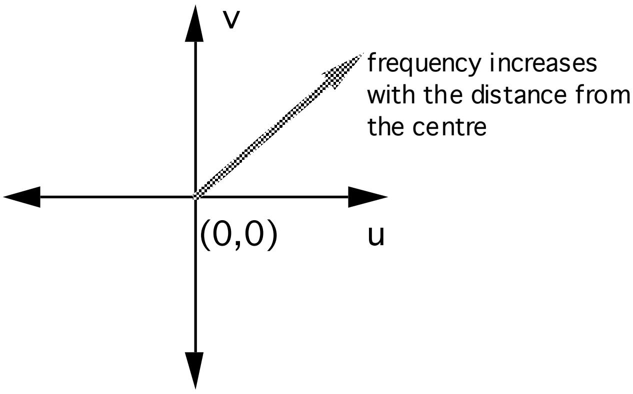
$$f(x) = \sum_u F(u) e^{j2\pi ux}$$

Interpretation: a particular point in the input signal can be represented by a weighted sum of complex exponentials (which are sinusoidal patterns) at different frequencies *u*. *F(u)* is a function which computes a weighting coeeficient for the different frequencies *u*.

Fourier transform for 2-dimensional images

Two-dimensional Fourier transform shows image frequencies in horizontal and vertical direction. The axes and values in the 2D frequency transform have the following interpretation:

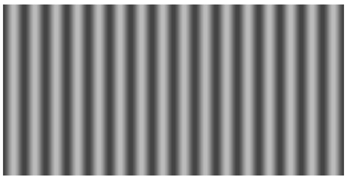
horizontal axis (u) - frequency component in horizontal direction
vertical axis (v) - frequency component in vertical direction
value at (u,v) - amplitude



Images below show simple image functions (on the left) and their 2-dimensional Fourier transforms (on the right). Because only horizontal variations are present in the image, only horizontal frequencies are showing in the Fourier transforms. As it can be seen, lower frequencies are near the centre, higher frequencies are located further from the centre.

sin(4x)

sin(8x)

sin(16x)  

Here is an example of Fourier transforms for a real image:

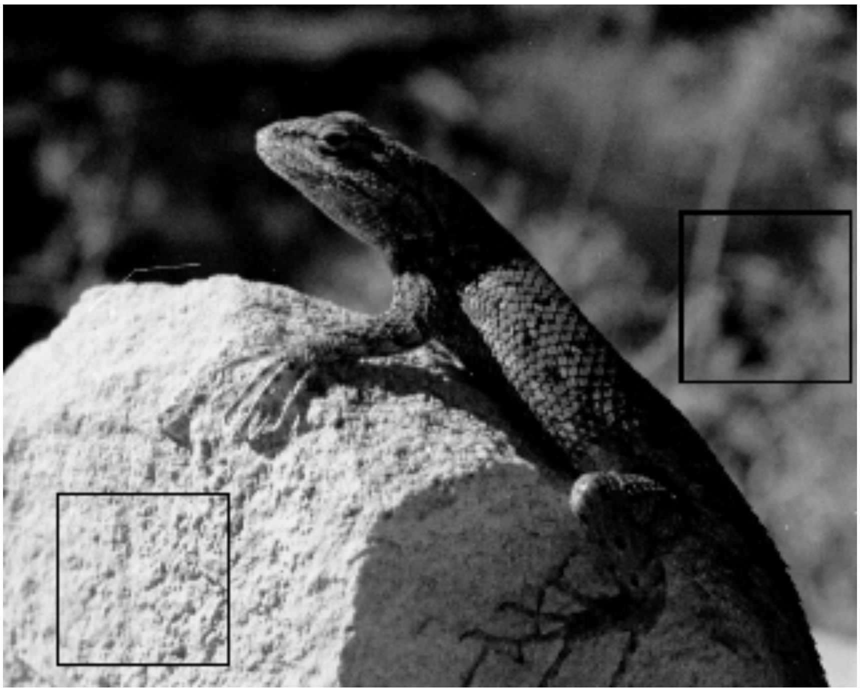
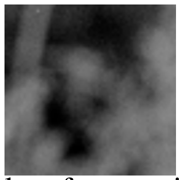
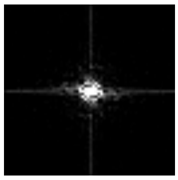
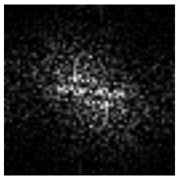


Image Transform

low frequencies

high frequencies

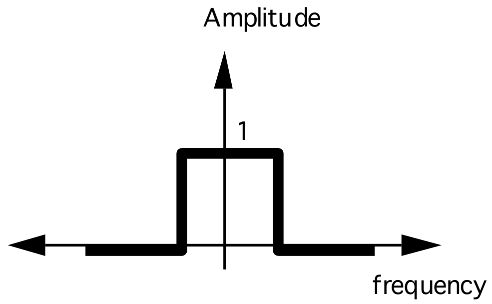
Having a frequency transform, where each frequency can be accessed explicitly via its (u,v) coordinates, frequencies can be directly manipulated, for example removed, suppressed or enhanced. A usual technique is to define a suitable *filter* in frequency domain and multiply the frequency transform by the filter. This operation is *convolution in frequency domain*. It is the exact equivalent of convolution in spatial domain which we have discussed earlier.

To perform convolution of an image with a filter in frequency domain:
- compute a Fourier transform of an image
- multiply by a filter function
- compute an inverse Fourier transform

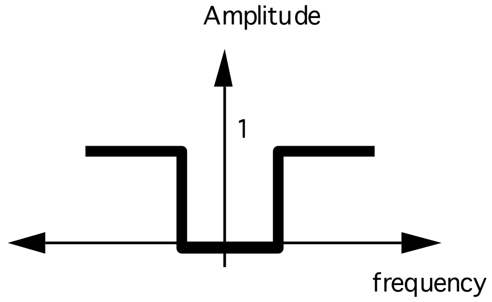
Filter functions

Let's look at 1-dimensional frequency filters first. Given a 1-dimensional frequency transform, multiply it with a desired frequency filter:

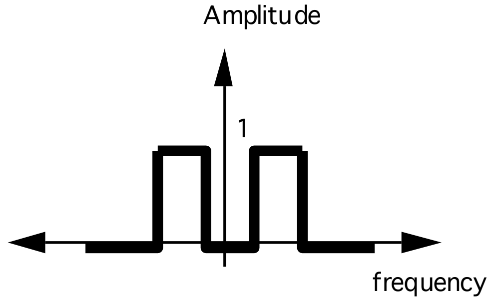
• Low frequency cut-off filter will keep low frequencies and remove high frequencies



• High frequency cut-off filter will keep high frequencies and remove low frequencies



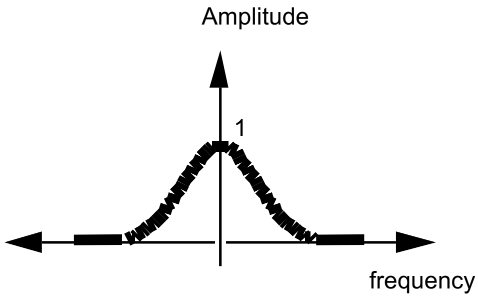
• Bandpass cut-off filter will keep only the selected range of frequencies and remove all other frequencies



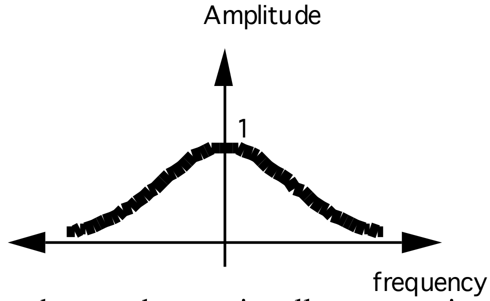
Two dimensional filters are simply created by "rotating" a 1-dimensional filter about the amplitude axis.

The filters shown above simply cut off the unwanted frequencies. Normally we want the transition between the wanted and unwanted frequencies to be gradual, so instead of sharp step-like transitions we use smooth transitions such as for example Gaussian:

• Low frequency cut-off filter will keep low frequencies and remove high frequencies



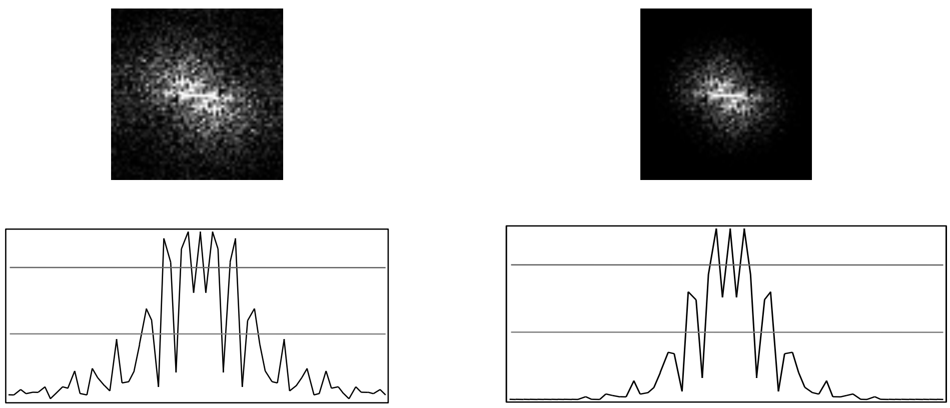
The filter above will remove less high frequencies than the one below and therefore the result will be less smoothed image.



Frequency filters in 2D do not have to be rotationally symmetric - an example of such asymmetric 2D Gaussian filter has been shown in handout 3 (Image enhancement).

Example of low-pass filter in frequency domain

Original Fourier transform and its image profile	Transform and image profile after convolution with a low-pass Gaussian filter



Frequency filtering operations separate frequency components within an image into different frequency ranges, most often into:

- low frequency components
- high frequency components

Following this separation, unwanted components can be rejected or, low and high frequency components can be processed separately and then re-combined.

Further reading and exploration

Bruce et al, Chapter 2 and Chapter 5
Sonka et al, Chapter 11
Umbaugh, Section 2.5
Watt, Chapter 4

Brigham EO (1988) The Fast Fourier Transform and its Applications: Prentice-Hall.

HIPR

Worksheeeets -> Digital Filters:

Frequency filters

CVIP

Analysis -> Transforms -> FFT

Walsh

Hadamard

Low pass / High pass

Band pass / Band reject

High frequency emphasis

Notch

View spectrum

Magnitude

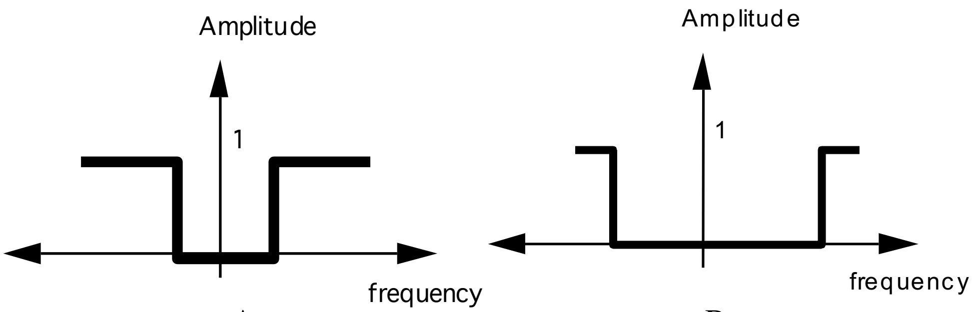
Log remapped

Phase

Exercises

1. Modify the shape of the bandpass filter shown above, so that the frequency cut-offs are gradual, not abrupt.

2. Which of the frequency filters, A or B, removes more of the high frequency components? Justify your answer.



A B

3. Observe, that all the frequency-domain filters shown above have values between 0 and why. Why?