SE3IA11/SEMIP12
Image Analysis

Image Enhancement in the Frequency Domain

Lecturer:
Prof. James Ferryman, Computational Vision Group
Email: j.m.ferryman@reading.ac.uk

Frequency Domain Enhancement

- The purpose of image enhancement is to emphasise some selected (desirable) image features and suppress others (undesirable features)
- In the last lecture, we examined enhancement in the spatial domain
- Several enhancement operations can be conveniently carried out in the frequency domain (where each image is represented by its Fourier transform)
- In this lecture we examine this topic, commencing with an introduction to image transforms and Fourier methods

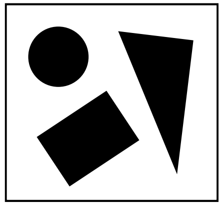
Image Transforms

- A digital image f(x,y) is normally represented as a 2D array of intensity values, with x & y being the horizontal and vertical indices – spatial domain representation
- Many image analysis tasks are made easier in other alternative representations

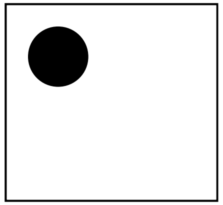
Image Decomposition

- How would you describe the content of the following hypothetical image?

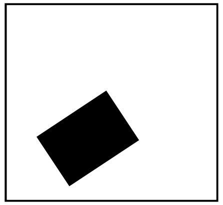
$f(x,y)$:



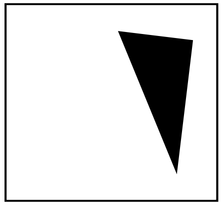
- If we have



$f_0(x,y)$:



$f_1(x,y)$:



$f_2(x,y)$:

then $f(x,y) = f_0(x,y) + f_1(x,y) + f_2(x,y)$, where $f_i(x,y)$, $i=0,1,2$, are used as reference or basis images (also called basis functions)

Hence the original image $f(x,y)$ is represented as a sum (linear combination) of three basis images

Image Transforms

- The above idea can be generalised to an arbitrary image f(x,y) and a pre-defined set of N basis images/functions $\{f_i(x,y); i=0,1,\ldots,N-1\}$

$$f(x,y) = \sum_{i=0}^{N-1} w_i f_i(x,y)$$

where w_i are the weighting coefficients which uniquely describe f(x,y) for a given set of basis images

Introduction to Fourier Analysis

- Important properties of processes are often made clearer by transforming the data to *represent the information* in another form
- As with many areas of IT, image analysis makes extensive use of the Fourier Transform (FT)
- The FT uses spatial frequencies (i.e. sinusoidal intensity distributions in space) to describe the image and the operations on it
- The importance of Fourier Analysis is due to the fact that separate sine-wave components of a signal can be analysed <u>individually</u>, to determine the performance of a linear system under <u>any</u> signal

Introduction to Fourier Analysis

- Knowledge of the response of a (linear) imaging system to the set of all possible sinusoids allows the output to be determined for *any* image
- The sinusoids "don't interact" under linear transformations, they form an *orthonormal basis set for additive signals*
- This is fairly familiar in acoustics – where we refer to sounds being high or low in frequency, and distinguish the smooth tones of a flute from the strident tones of a violin
- Both instruments may play the same *fundamental note*, but differ in their *harmonic components*
- Likewise, we expect a good quality amplifier to transmit pure tones without adding unwanted *distortion* components, thus allowing both the violin and the flute to remain identifiable
- The same sine-wave analysis can be applied to 2D images

Convolution

- Assume F(u,v), P(u,v) and Q(u,v) are the FTs of images f(x,y), p(x,y) and q(x,y)
- If F(u,v) = P(u,v)Q(u,v), what is f(x,y) in terms of p(x,y) and q(x,y)?
- Someone has proved for you that it's given by:

$$f(x,y) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} p(k,l)q(x-k,y-l)$$

- Calculation of the above forms appears frequently in signal and image analysis, so it's written more compactly as

$$f(x,y) = p(x,y)*q(x,y)$$

- The above calculation is called *convolution*. Convolution is commutative, i.e., p*q=q*p. When computing p*q or q*p, p & q are assumed to be periodic functions to cope with negative indices, i.e., p(k,l) = p(N+l, N+l), etc.

Illustrating the Convolution Theorem

- Consider an input image i(x,y), which is blurred by a weighting function b(x,y), to form an output o(x,y).
- We have $o = i \otimes b$ (space domain) or $O = I * B$ (frequency domain)
- (x,y) are space domain parameters, (ξ, ψ) are frequencies.

		Fourier Transformation	
Input image	i(x,y)	$\Leftrightarrow$	$I(\xi,\psi)$
	$\otimes$		$*$
System properties	b(x,y)	$\Leftrightarrow$	$B(\xi,\psi)$
	$\Downarrow$		$\Downarrow$
Output image	o(x,y)	$\Leftrightarrow$	$O(\xi,\psi)$
	Space Domain		**Fourier Domain**

Illustrating the Convolution Theorem

- Notes:
  - The Blur function, b(x,y), is often called the Point Spread Function (PSF) – it measures how an infinitesimal point (also called a delta function) is *spread* out by the system
  - Its spectrum, $B(\xi,\psi)$, is often called the Modulation Transfer Function (MTF) – it measures how the modulation of each component is *transferred* (i.e. amplified or attenuated) by the system
  - Either the PSF or MTF provide a *complete description* of a linear shift-invariant system

Convolution

- In summary, we have

  multiplication (convolution) in freq domain

  $\Leftrightarrow$

  convolution (multiplication) in spatial domain

  This is called the *convolution theorem*. It often allows complicated processing (e.g. filtering) in the spatial domain to be performed very efficiently in the frequency domain

Fourier Properties

- A summary of some important properties of Fourier Analysis:
  - Any image can be represented as the sum of a set of sine and cosine distributions – this is more neatly expressed as a complex exponential, which conveniently combines the cos and sin terms

    Thus: $$F(\xi,\psi) = \frac{1}{\sqrt{2\pi}} \sum f(x,y) e^{-j(\xi x + \psi y)}$$

    Any function is identically equal to the sum of its harmonics

  - Reciprocal analysis & synthesis: if $F(\xi,\psi)$ is the transform of f(x,y), then f(x,y) is also the transform of $F(\xi,\psi)$; they are said to be Fourier mates (or pairs).

Fourier Properties

- A summary of some important properties of Fourier Analysis:
  - <u>The Convolution Theorem</u>: $h = f \otimes g$ (each a function of (x,y)) can be expressed very easily in the Fourier Domain.

    The Fourier Transform of *h* is the *product* of the Fourier Transforms of the separate functions: $H = F^*G$ (each a function of (ξ, ψ)).

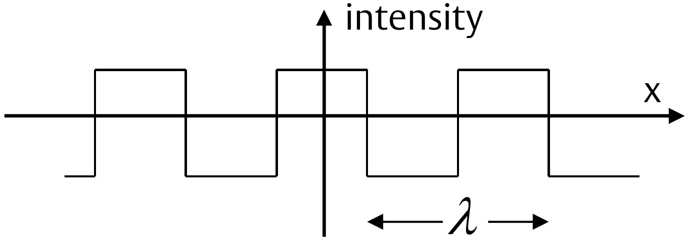
    <u>Hence: under any convolution, the harmonics only ever change their amplitudes, and do not change frequencies (i.e. there is no distortion).</u>

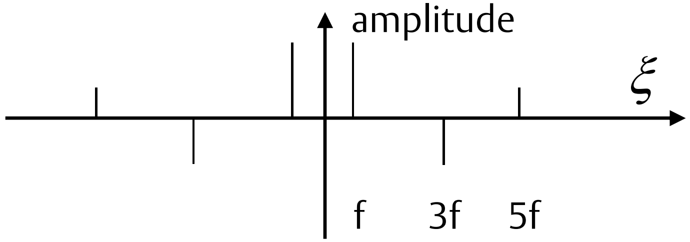
  - <u>Reciprocal size relationship</u> between the two domains: large objects in space, are small in frequency space & vice-versa.
    - This is due to the fact that frequencies represent cycles *per* space metric

  - (For Mathematicians only!) Complex exponentials are the *eigenvectors* of the class of linear transformations, they always transform into scalar products of themselves. The *eigenvalues* (the scalars, which form the system *spectrum*) completely characterise the transformation

Spectral Analysis

- Return to considering convolution operators
- Both the *input image* (the signal) and the convolution *weighting function* (the operator system response) can be represented as frequency distributions, i.e. their Fourier Transforms
- A very simple example: the square wave (in 1D):





- Any periodic distribution, of period λ , transforms into a discrete series of Fourier components: the *fundamental* (f = $1/\lambda$), and all integral multiples, 2f, 3f, 4f, …
- In the case of a *square wave*, the values of all even harmonics is 0 and only the odd harmonics are present – (ξ = f, 3f, 45, 7f, …). Their amplitudes decay as the inverse of frequency: $4a/\pi$, $4a/3\pi$, $4a/5\pi$, … where a is the amplitude of the square wave, i.e. (max-min)/2

Spectral Analysis

- "Blurring" a square wave, by means of a linear, shift-invariant operator, merely changes the amplitudes of the components
  - It does not introduce any new components

  [Amplitudes are multiplied by the amplitude of the weighting function at the frequency]

- We can also go the other way: an image can by synthesised simply by adding up the values of a set of components at each point. This is expressed mathematically as the inverse Fourier Transform:

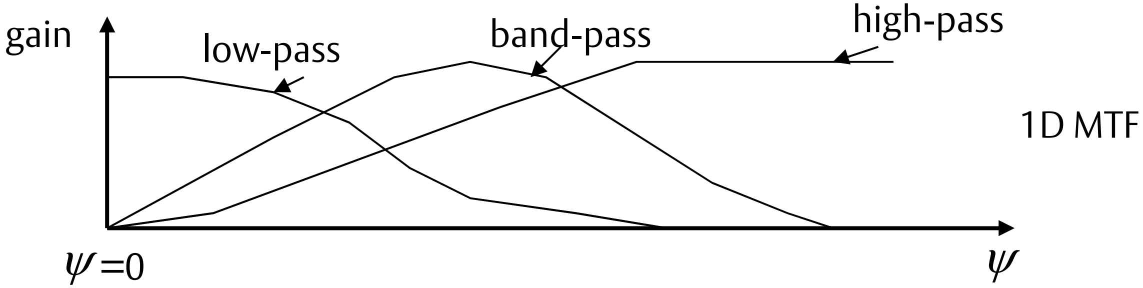
$$f(x,y) = \frac{1}{\sqrt{2\pi}} \sum F(\xi,\psi) e^{j(\xi x + \psi y)}$$

  for all (ξ,ψ)

- Note the near symmetry of the forward and inverse transforms – only the Fourier pairs and the sign of the complex exponential changes

Fourier Jargon

- Fourier methods provide an *alternative description* of the information content of a signal, and jargon from radio theory has been adopted
- Filtering processes are often said to be low-, high-, or band- pass according to the gain at different frequencies
- Band & High-pass filters are said to be AC-coupled, and have zero response (i.e. the value at $\psi=0$ is 0)
- Low-pass filters transmit the steady signal (the mean over time or space), and are said to be DC-coupled



- The finest detail in an image is carried by its highest frequencies. Low-pass filtering therefore corresponds to smoothing, which suppresses the fine detail, but has little effect on the low frequencies

Discrete Fourier Transform (DFT)

- DFT is the sampled Fourier Transform
  - When dealing with digital images, we are never given a continuous function
    - Must work with finite number of discrete samples
    - Samples are pixels that compose an image
    - Analysis of images requires the DFT
  - DFT does not contain all frequencies forming an image; instead contains a set of samples sufficient enough to fully describe the spatial domain image
  - Number of frequencies corresponds to number of image pixels
    - i.e. image in spatial and Fourier domain are of same size

Discrete Fourier Transform (DFT)

- The general forms of image transforms

Forward: $$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v)$$

Inverse: $$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)h(x,y,u,v)$$

In DFT both transformation kernels (g & h) are complex functions, and the results are in general also complex.

If the kernels are real functions of the following form

$$g(x,y,u,v) = h(x,y,u,v) =$$
$$\frac{2c(u)c(v)}{N}\cos\left(\frac{2x+1}{2N}u\pi\right)\cos\left(\frac{2y+1}{2N}v\pi\right)$$

where c(0) = 1/√2 and c(w)=1 for w=1,2,…,N-1, then the transform is called the discrete cosine transform or DCT (because kernels are cosines).

Note that the cosines are also separable in x & u and y & v, so a 2D DCT can be obtained from two 1D DCTs as in DFT

The DCT kernels may be written in terms of DFT kernels. This allows fast DCT via FFT.

Discrete Fourier Transform (DFT)

- For a square image of size NxN, let the basis set be defined by the following N^2 complex sinusoidal functions:

$$h(x,y,u,v) = \frac{1}{N}\exp\left[\frac{j2\pi}{N}(ux+vy)\right]; u,v = 0,\ldots,N-1$$

then we can represent an image f(x,y) as a weighted sum of these sinusoidal functions:

$$\begin{aligned} f(x,y) = {} & F(0,0)h(x,y,0,0) + F(0,1)h(x,y,0,1) \\ & + F(0,2)h(x,y,0,2) + \ldots + \\ & F(N-1,N-1)h(x,y,N-1,N-1) \end{aligned}$$

or more compactly

Discrete Fourier Transform (DFT)

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)h(x,y,u,v)$$

or

$$f(x,y) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\exp\left[\frac{j2\pi}{N}(ux+vy)\right] \quad (1)$$

Interestingly, the weighting coefficients F(u,v) are calculated from the original image in a similar equation:

$$F(u,v) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(x,y)\exp\left[\frac{-j2\pi}{N}(ux+vy)\right] \quad (2)$$

Discrete Fourier Transform (DFT)

Let

$$g(x,y,u,v) = \frac{1}{N}\exp\left[\frac{-j2\pi}{N}(ux+vy)\right]$$

then (2) may be equivalently written as

$$F(u,v) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(x,y)g(x,y,u,v)$$

Equations (1) and (2) are said to define a Fourier transform (DFT) pair, where (2) is the forward transformation (i.e. to map image to Fourier domain), and g(x,y,u,v) the forward transformation kernel, and (1) the inverse transformation (i.e. to map from Fourier to spatial domain) and h(x,y,u,v) the inverse transformation kernel. Since u & v are related to the frequencies of the sinusoidal functions, they are called frequency indices

- Note F(u,v) calculated in (2) is in general a complex value, i.e., it has a real part $F_R(u,v)$ and an imaginary part $F_I(u,v)$:

$$F(u,v) = F_R(u,v) + jF_I(u,v)$$

Discrete Fourier Transform (DCT)

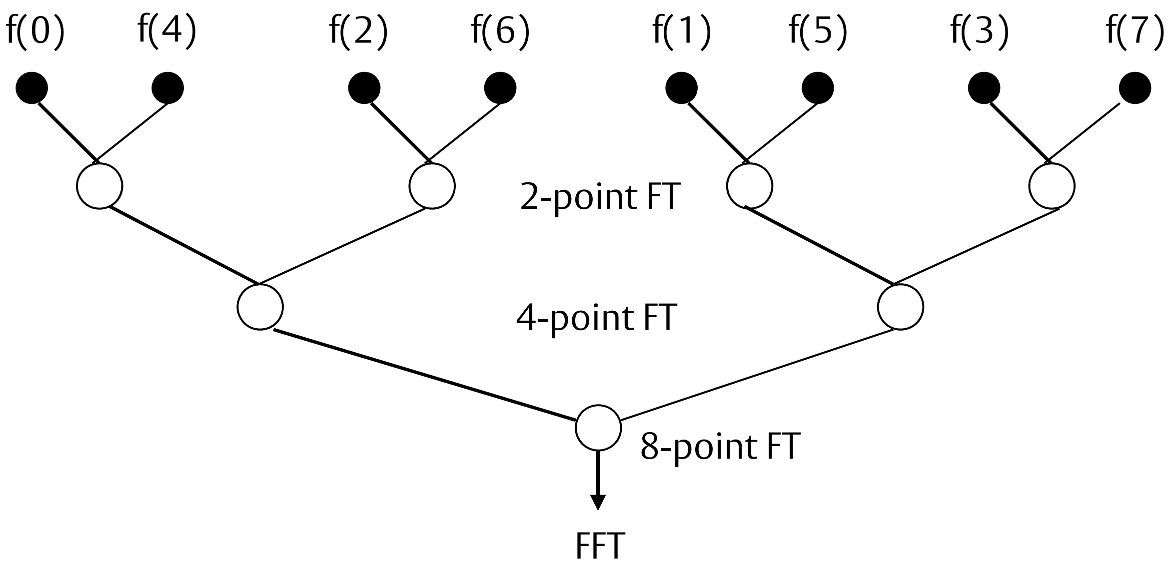
- FT produces a complex number valued output image
  - can be displayed as two images
    - Either *real* and *imaginary* part or *magnitude* and *phase*
- In image analysis, often only magnitude of FT is displayed
  - contains most of information of geometric structure of spatial domain image
- However, if one needs to re-transform the Fourier image into the correct spatial domain after some processing in the frequency domain, must ensure *both magnitude and phase of Fourier image are preserved*

Computational Methods

- The Fast Fourier Transform
- Direct calculation of the 1D FT of a N-point function f(x) requires multiplications in the order of N^2, a slow process for large Ns
- Efficient method discovered in 1942, but not applied until later due to lack of computing power

Fast Fourier Transform (FFT)

- It was found that a N-point FT can be obtained from two N/2-point DFTs each of which can in turn be obtained from two N/4-point DFTs and so on – the *successive doubling* method (see below)
- Coined the Fast Fourier Transform (FFT) (Cooley & Tukey, 1965)
- FFT reduces number of complex multiplications from N^2 to order of $N \log_2 N$



Fast Fourier Transform (FFT)

- The divide and conquer technique used allows for significant speedup over DFT for 2D images

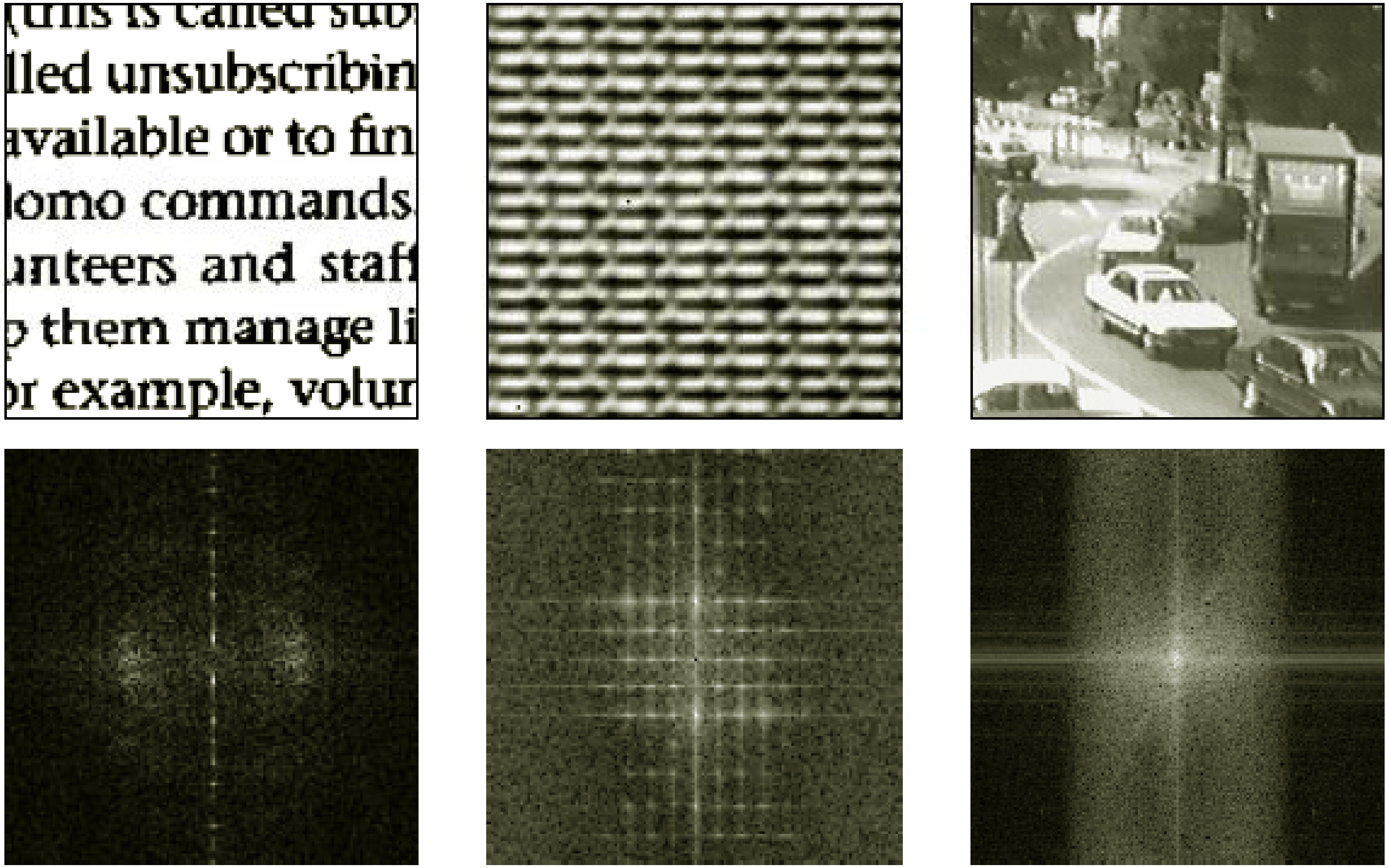
Image Size	DFT Multiplications	DFT Time	FFT Multiplications	FFT Time
256 x 256	4.3 E9	71 min	1,048,576	1.0 sec
512 x 512	6.8 E10	19 hr	4,718,592	4.8 sec
1024 x 1024	1.1 E12	12 days	20,971,520	21.0 sec
2048 x 2048	1.8 E13	203 days	92,274,688	92.2 sec

Savings when using FFT on 2D image data

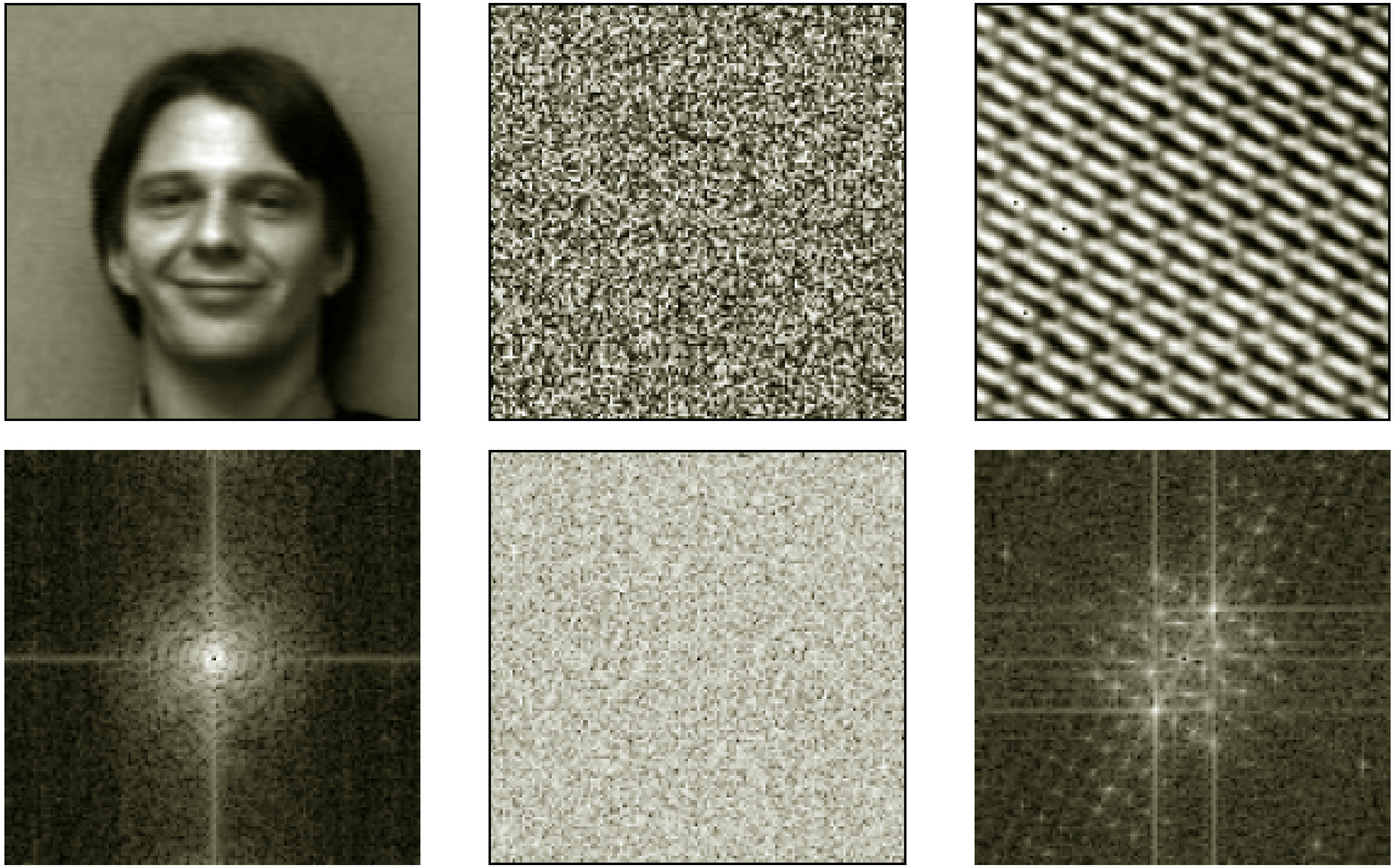
Fast Fourier Transform (FFT)

- For an efficient implementation of the FFT in C, consult the book *Numerical Recipes in C*, available on the Web at www.nr.com
  - *http://www.nrbook.com/a/bookcpdf.php*
  - *Chapter 13: Fourier and Spectral Applications*

Examples of Fourier Spectra



Examples of Fourier Spectra



Significance of Spectral Peaks

- Peaks in the frequency domain indicate global periodic structures in the spatial domains
- Direction (angle with u-axis) of spectral peaks is direction (angle with x-axis) of corresponding global structure +/- 90^0

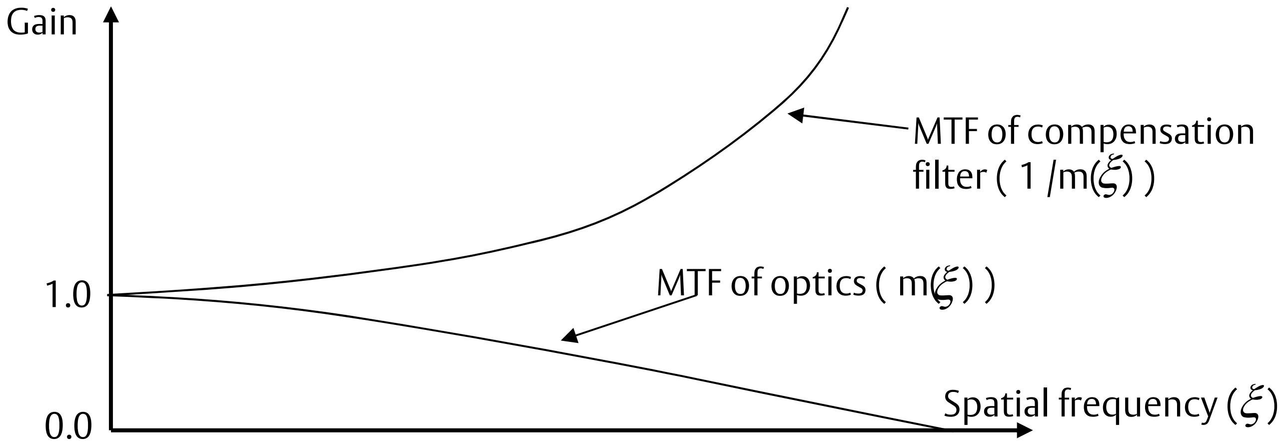
Fourier Transform – Guidelines for Use

- FT is used to access the geometric characteristics of spatial domain image
- Since image in FD is decomposed into its sinusoidal components, easy to examine or process certain image frequencies, influencing geometric structure in spatial domain
- In most implementations, Fourier image is shifted such that DC-value (i.e. image mean) F(0,0) is displayed in centre of image
  - i.e. further away from centre an image point is, the higher its corresponding frequency
- Phase information is crucial to reconstruct image in spatial domain

Application: Image Reconstruction

- One of the simplest applications of Fourier Methods in image analysis is to correct the image for blur due to optical defects
- A typical lens. or any form of electronic sensor, imposes a degree of smear on an image, due to poor optics or the finite size of the sensor element
- The MTF of a system, $m(\xi)$ can be measured experimentally – simply present the system with a sine-wave input, and measure the amplitude (and phase) of the output
- The MTF of an imperfect system commonly takes the form of a low-pass filter which attenuates the high frequencies progressively, until the cut-off frequency, where $m(\xi) = 0$, and the information is lost

Application: Image Reconstruction



- But it is perfectly feasible to define a <u>compensation filter</u> whose gain at each frequency is exactly 1/m(ξ)
- Consider what happens if we convolve the (blurred) measured image with the new filter
- Convolution is multiplication in the Fourier domain, so each component of the input is first attenuated by the blur, but then amplified back to its original level
- This happens at each frequency, so the result (in theory) is a perfectly reconstructed image, with all the blur removed

Application: Repetitive Signal Removal

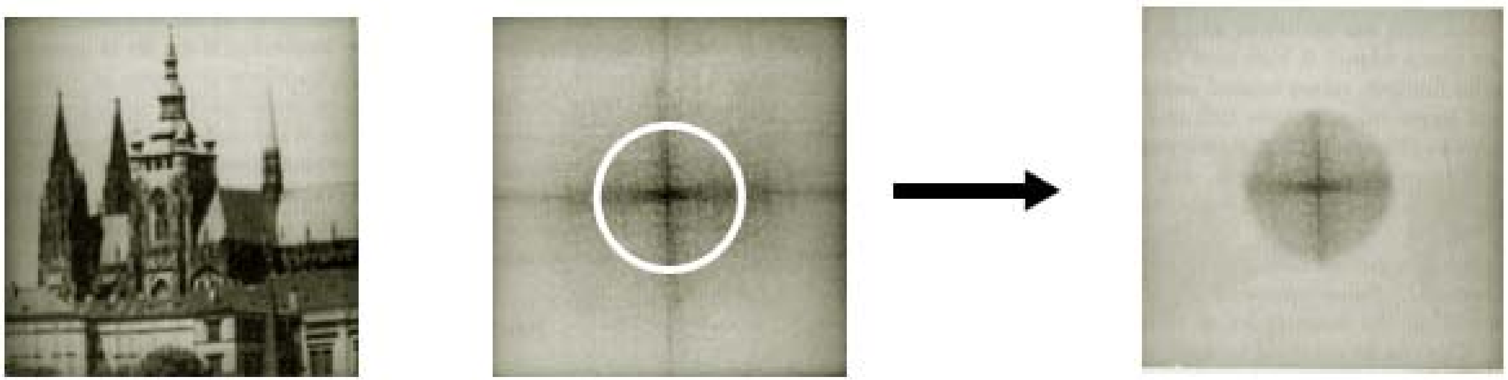
- Another example of Fourier methods in image analysis is to remove unwanted repetitive signals – e.g. eliminating video lines by <u>blocking</u> the (vertical) frequencies which they introduce
- Note the similarity to "notch filters", which remove hum, or the modulation frequency in audio samples
- These methods have often been implemented optically, since the processing can be done at the speed of light!
- Their digital equivalents are also widely used, since although slower, they are far more flexible

Application: Filtering

- Filtering is one operation whereby some image components are removed (i.e. filtered out) so as to enhance others

Lowpass Filtering

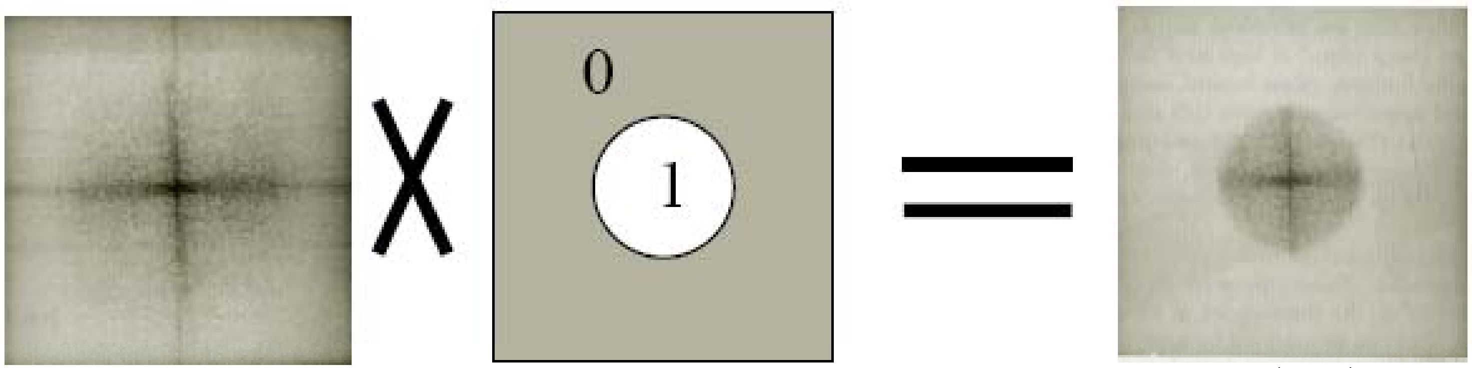
Let's say we wish to remove HF components in order to enhance the LF ones. We can do this by setting the value of the Fourier spectrum at HF locations to zero and leave the LF values unchanged. The inverse FT of the modified frequency domain (FD) representation gives the desired image



Lowpass Filtering

- The above zero-setting operation is equivalent to multiplying the original FD representation by a function H(u,v) defined as

$$H(u,v) = \begin{cases} 1; & \text{for LF locations (e.g. within a circle)} \\ 0; & \text{for HF locations (e.g. outside the circle)} \end{cases}$$

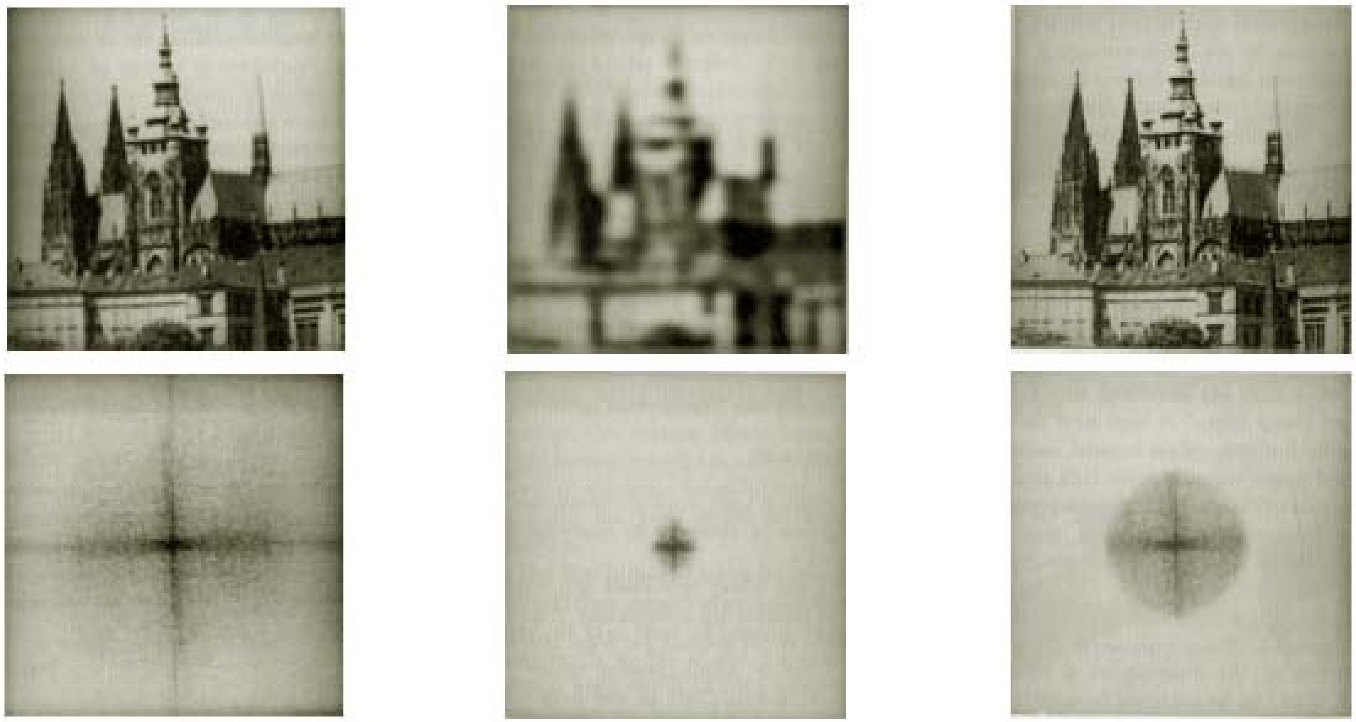


P(u,v) H(u,v) Q(u,v)

$$Q(u,v) = P(u,v)H(u,v)$$

Lowpass Filtering

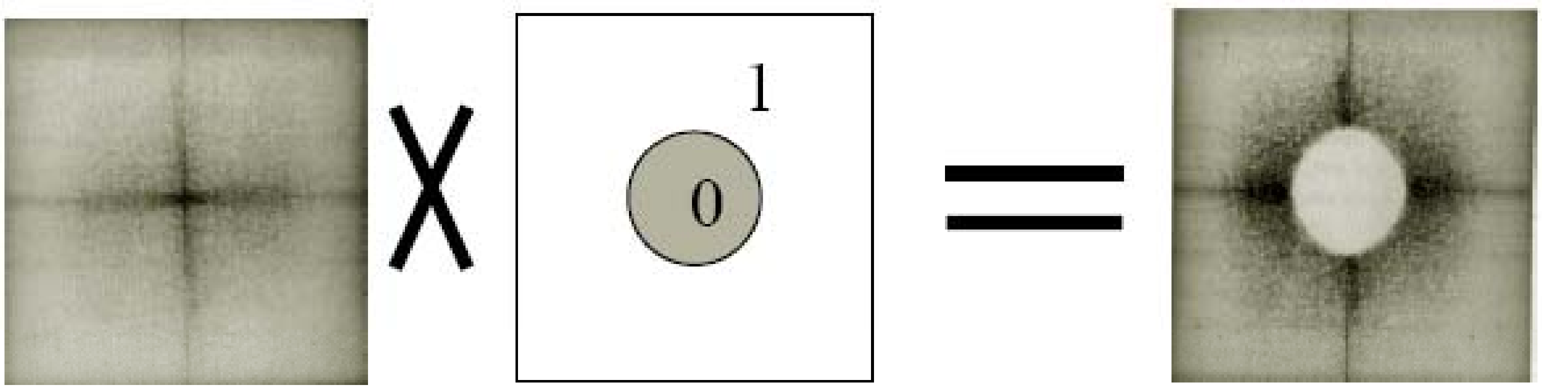
- Since H(u,v) serves as a filter which only lets LF components pass, it's called a low-pass filter (LPF). Its shape doesn't have to be circular. Further examples of lowpass filtering are given below:



Highpass Filtering

- This is performed similarly. In this case, the filter function H(u,v) keeps HF values unchanged and erase all LF ones, e.g.:

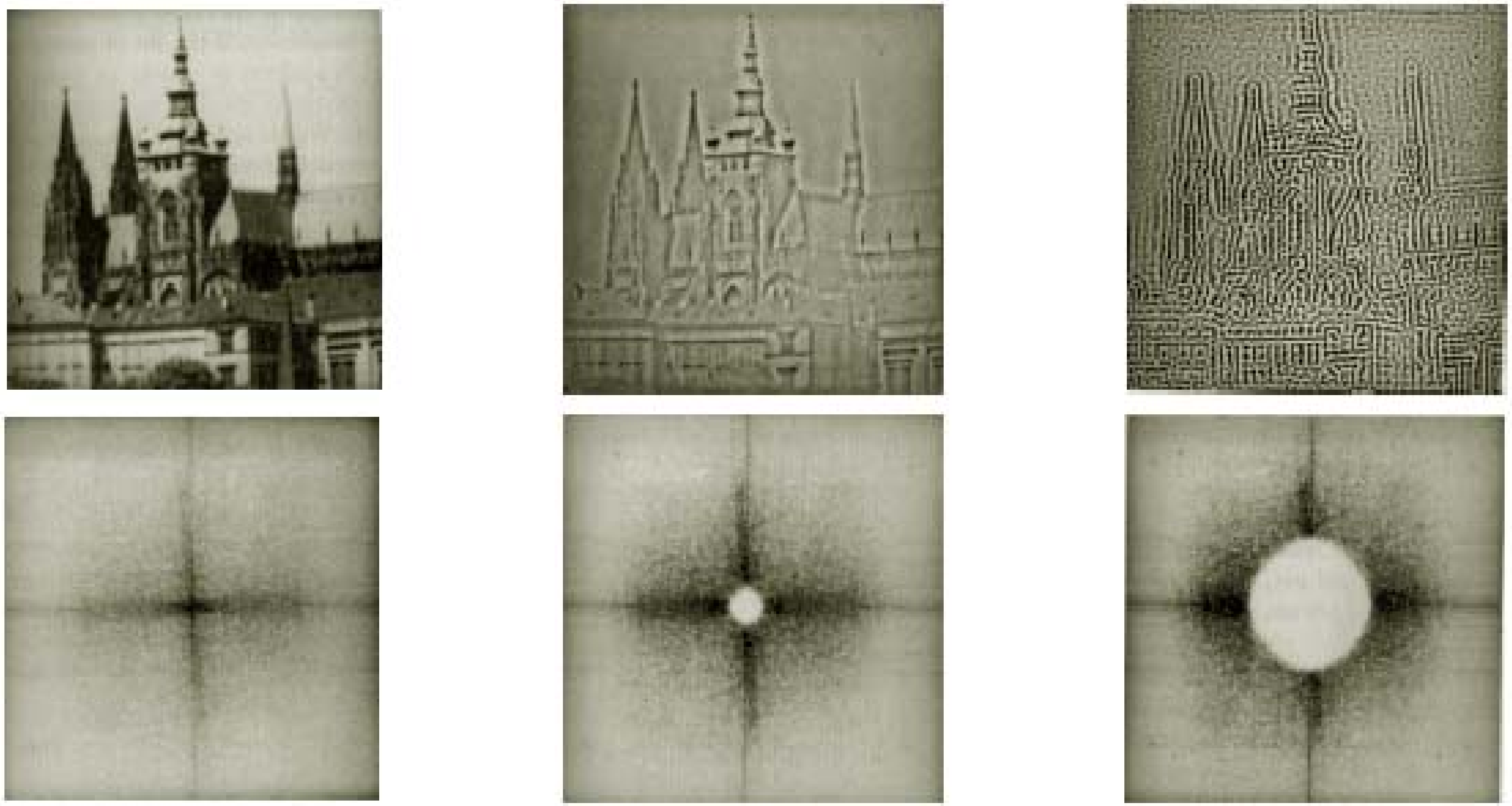
$$H(u,v) = \begin{cases} 0; & \text{If (u,v) is within a circle} \\ 1; & \text{otherwise} \end{cases}$$



P(u,v) H(u,v) Q(u,v)

Q(u,v) = P(u,v)H(u,v)

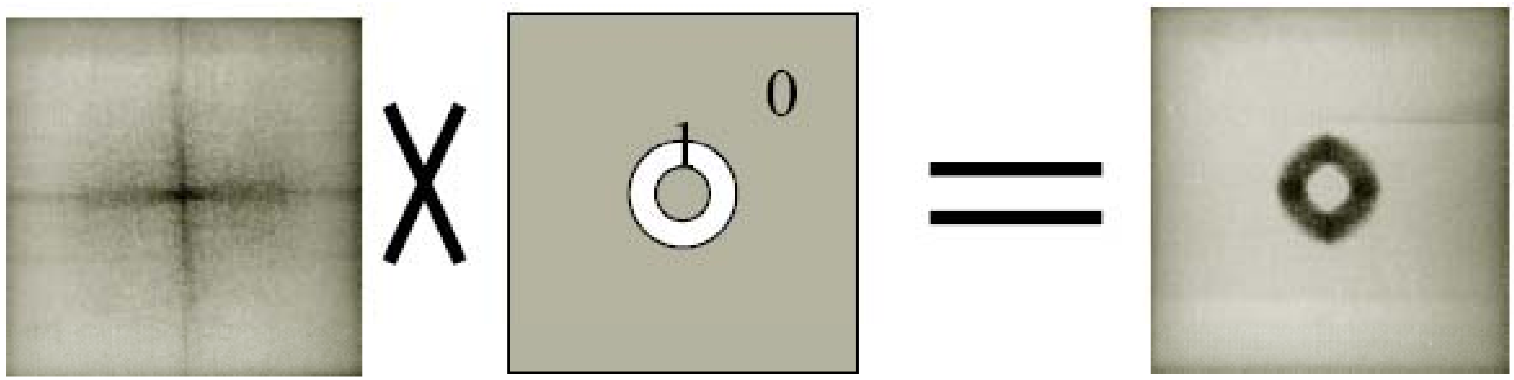
Examples of Highpass Filtering



Bandpass Filtering

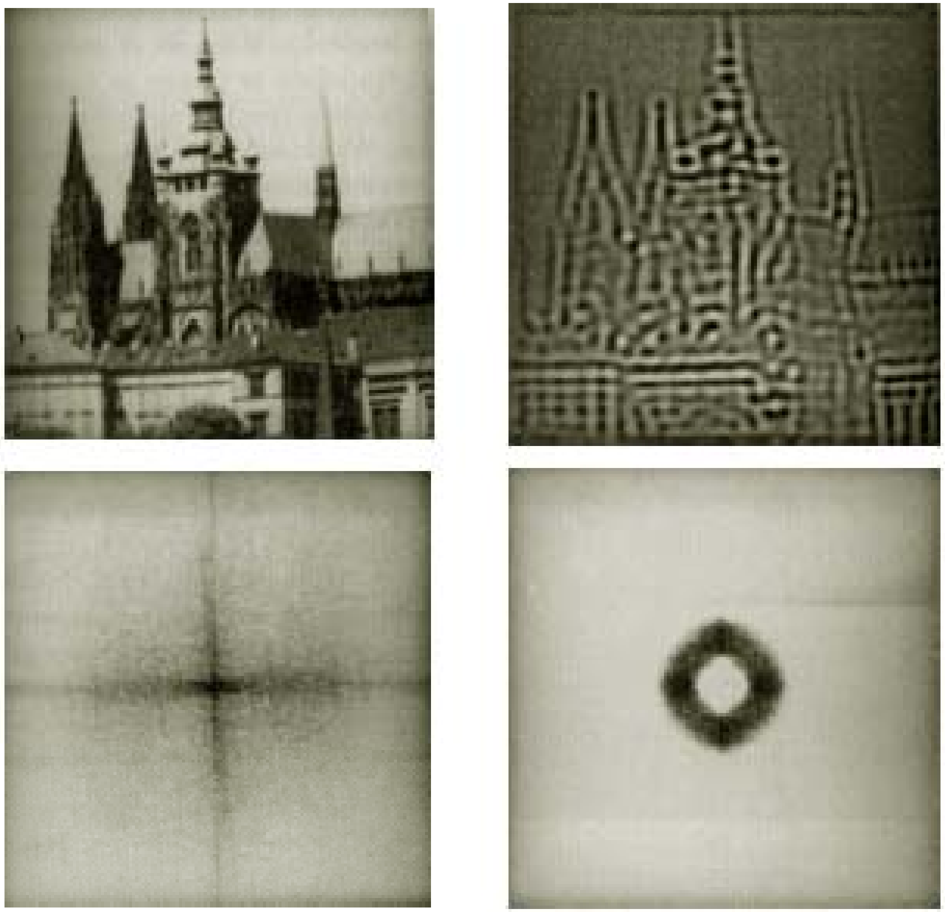
- Only keeps values corresponding to frequencies in a given interval, so a bandpass filter (BPF) may be defined as

$$H(u,v) = \begin{cases} 1; & \text{If (u,v) is in overlaps of two circles} \\ 0; & \text{otherwise} \end{cases}$$



P(u,v) H(u,v) Q(u,v)

Example of Bandpass Filtering



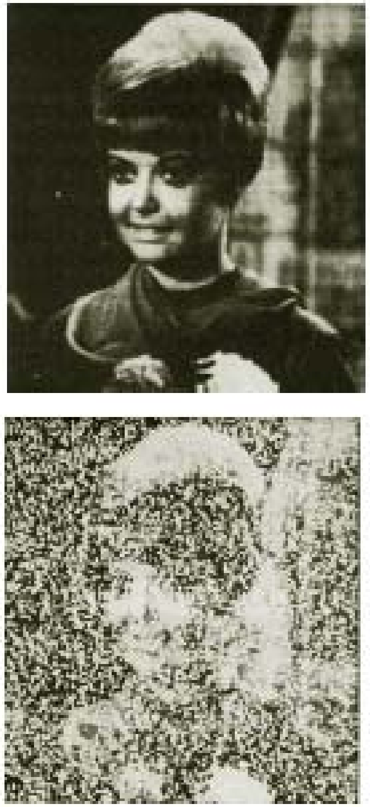
Application: Noise Removal

- Intensity values of pixels do not always reflect the true information about the underlying scene (i.e. the object photographed) since these values may also have been altered undesirably (e.g. due to dirt on the scanner, camera, etc.) If so, the image is said to be *noisy* or have been contaminated by noise

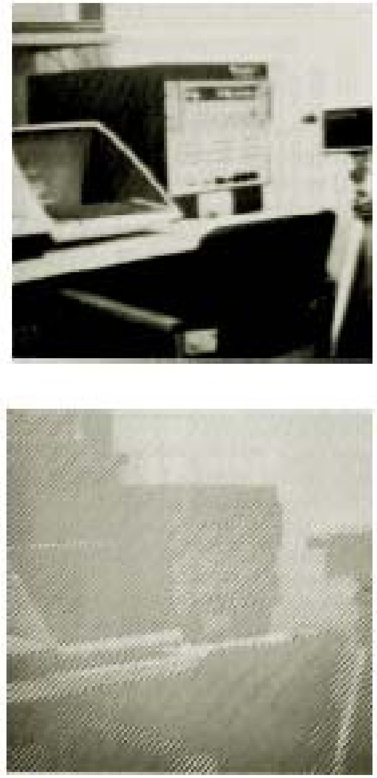
- If the measured intensity value is a sum of the true value and a small term due to noise, the image contains *additive noise*
- If the intensity of some of the pixels is predominantly due to noise and is significantly different from those of neighbouring pixels, the image contains *impulse noise* (or salt-and-pepper noise)
- If the noise term forms some sort of structure, the image contains *structured noise* (or coherent noise)

Examples of Noisy Images

Additive



Impulse



Structured

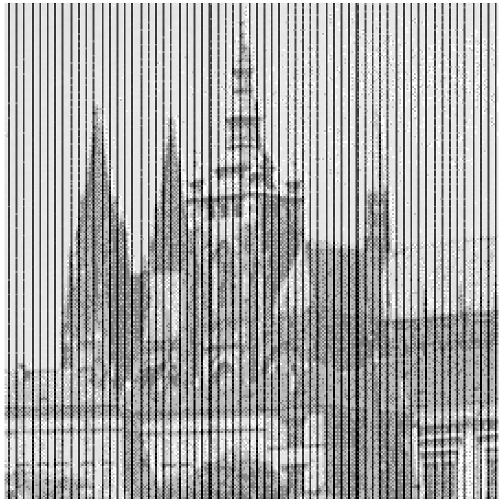
Lowpass Filtering

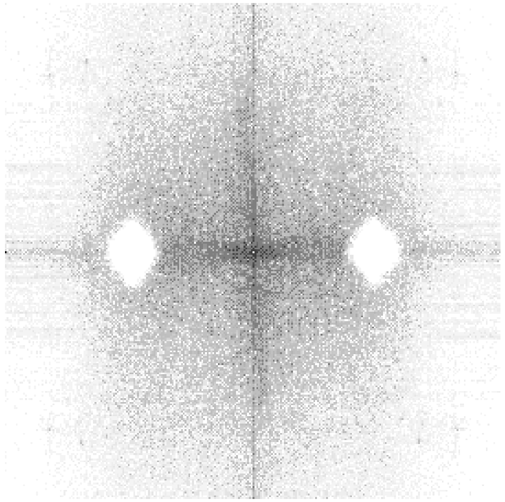
- Natural images usually contain relatively low frequency components and significant HF components are likely due to noise
- Hence a large part of noise may be removed by means of lowpass filtering
- This can be done by using a LPF in the FD, or alternatively by local averaging (see last lecture.)

Lowpass Filtering: Noise Removal

- A problem with noise removal via LPF is the loss of sharpness as edges are blurred
- Can be solved via e.g. Median Filtering (see last lecture)

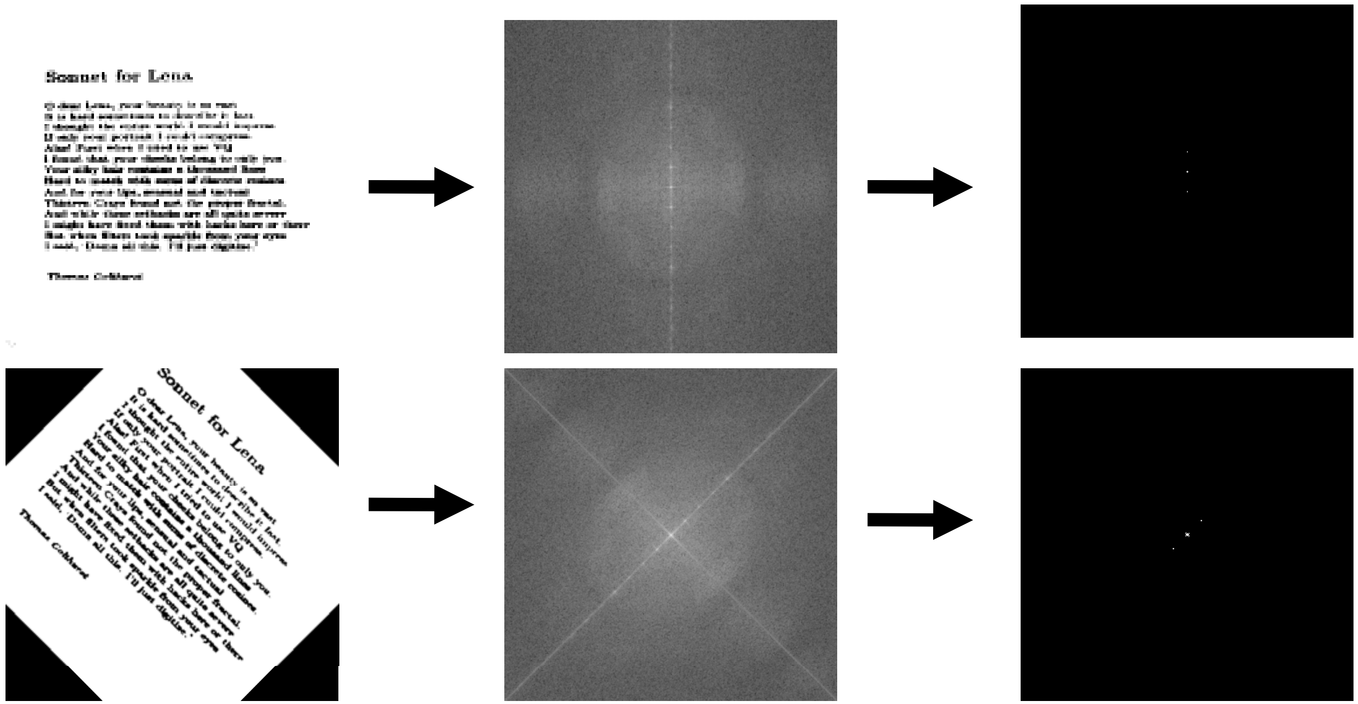
Example of Periodic Noise Removal





Application: Text Orientation

- Text recognition using image analysis techniques is simplified if we assume that text lines are in a predefined direction
- FT can be used to locate initial orientation of text
- A rotation can then be applied to correct error



Other Transforms

- The FT is not the only transform used in image analysis
- Other transforms include:
  - Hilbert, Hartley, Hough, Hotelling, Hadamard, Haar, Walsh, Wavelet, Karhunen-Loève
  - As well as the DCT, DST
- However, the FT, due to its wide application base, is one of the most popular
- The next slide summarises the Wavelet Transform
- We then examine some applications of the FT to image enhancement

Wavelet Transform

- The DFT *is good* for analysing global image features such as periodicity of certain image structures
- However, DFT is *not useful* in describing more local features (e.g. frequency components in small image region)
- Can be overcome by applying DFT to small image regions at various locations (called *windowed Fourier Transform*)
- The solution appears to be obvious: use a set of sizes (scales) and a set of locations (to enable you to see both the forest and trees, so to speak.)
- The above is the basic idea behind the Wavelet Transform, where the basis functions are the scaled (dilated) and translated versions of a "mother" function or wavelet, e.g.:
- Mother wavelet: g(x)

  Other wavelets: $g_{a,b}(x) = \frac{1}{\sqrt{a}} g\left(\frac{x-b}{a}\right)$

  Most research has focussed on the choice of the Mother wavelet

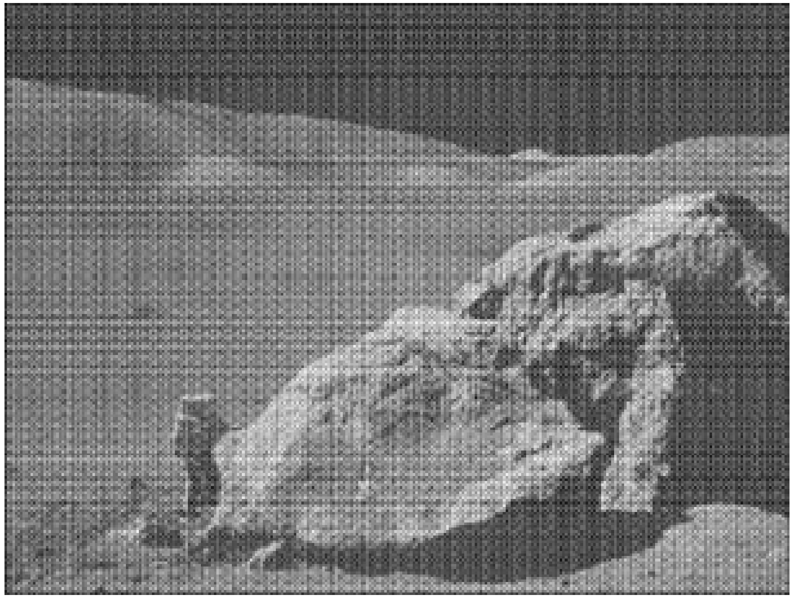
Case Study: Customised Digital Filters

- The following provides a case of study of customised digital filtering applied to noisy image data

Why Image Enhancement?

- Noise/ Distortion reduction: preprocessing step before image segmentation
- Easiest way: blind smoothing using a spatial LP filter
- BUT: still regular unwanted patterns

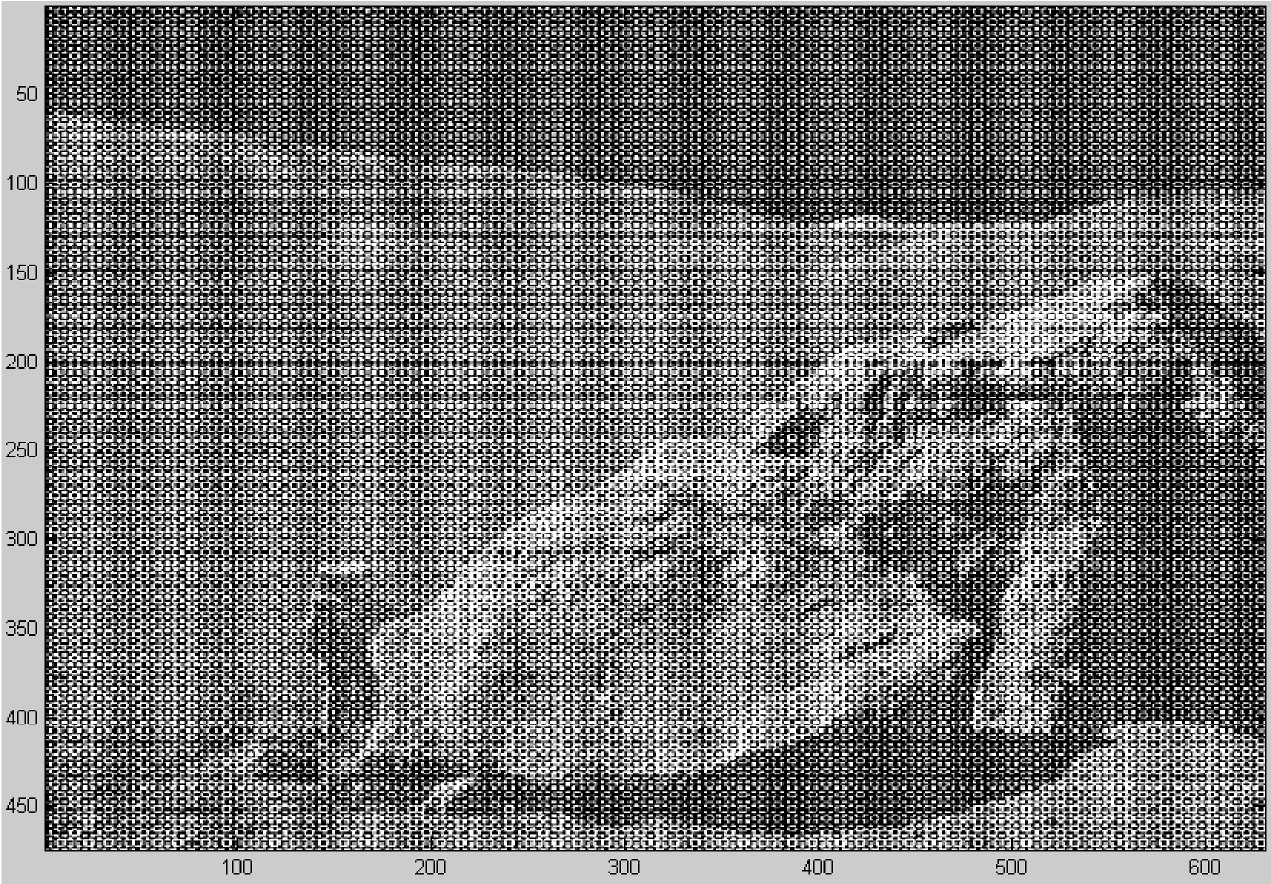
$$mask = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



The Algorithm

1. Perform a 2D-FFT and centre the spectrum of the corrupted image
2. Estimate disturbing frequency by finding the maximum energy
3. Design a band reject filter according to the estimated frequency
4. Filter corrupted image with band reject filter
5. Perform a 2D-IFFT to obtain the enhanced image

The Original Distorted Image f(x,y) from the Moon's Surface

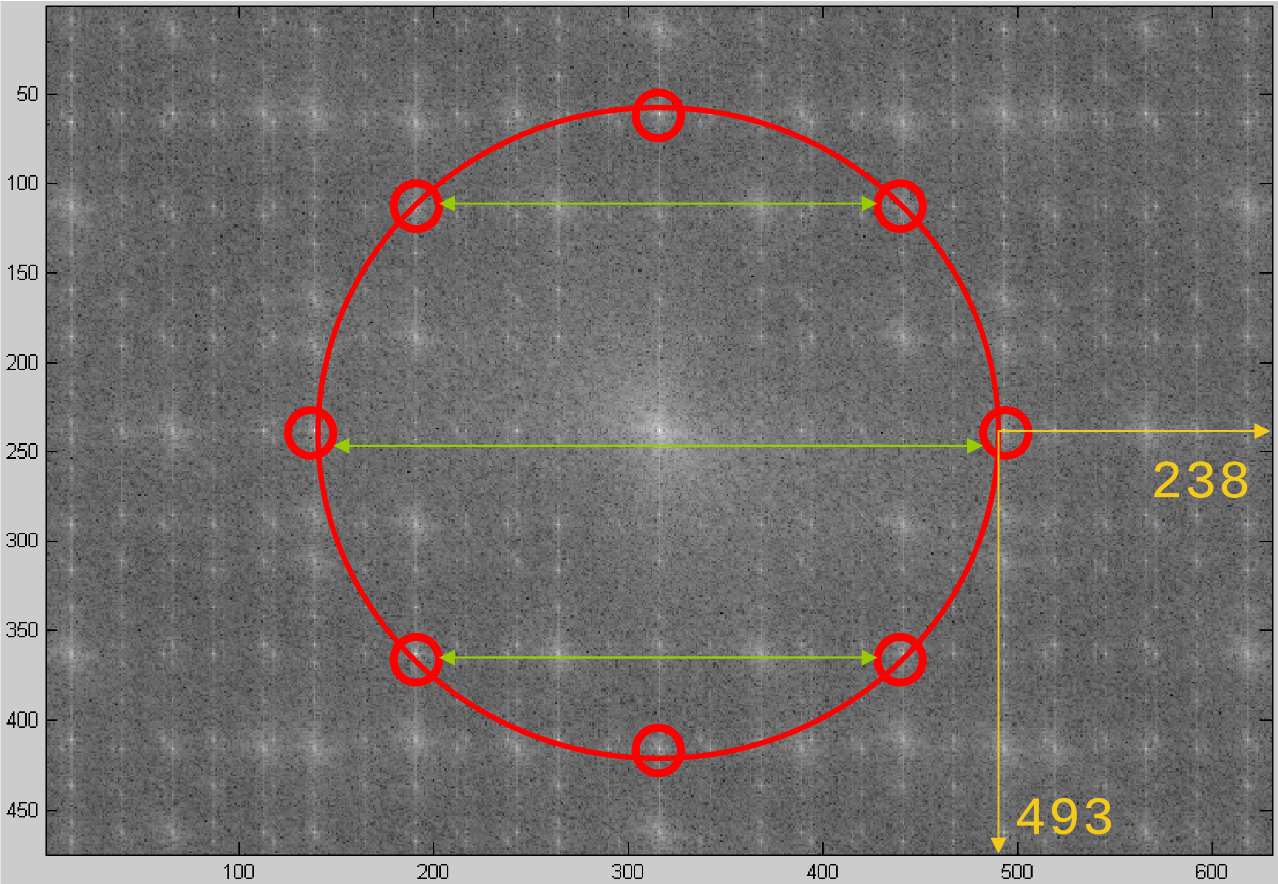


- <u>Periodic Noise</u>

- Frequent signal interference during image acquisition and / or transmisssion

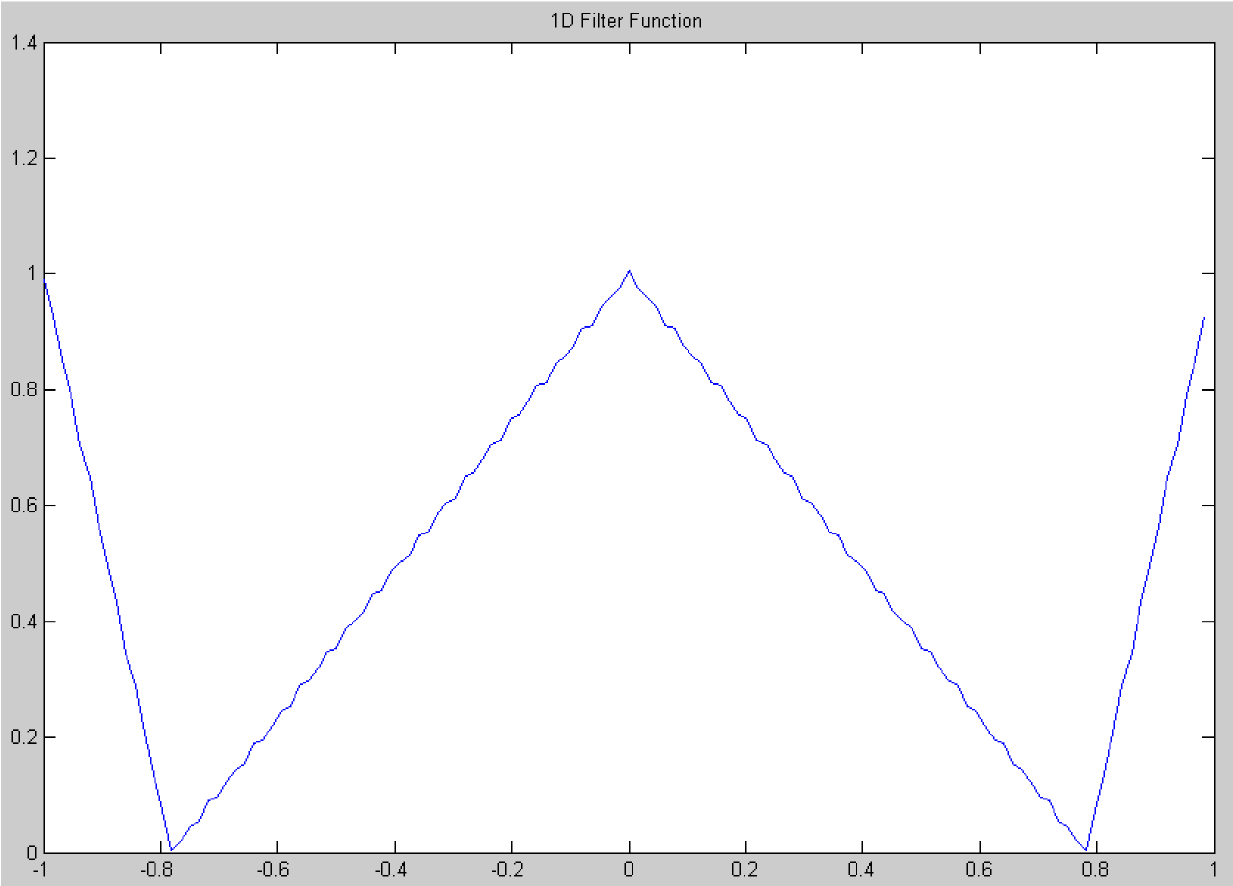
- Human eye is tolerant enough, whereas edge detection algorithms will fail

The Original Image's Spectrum F(x,y)



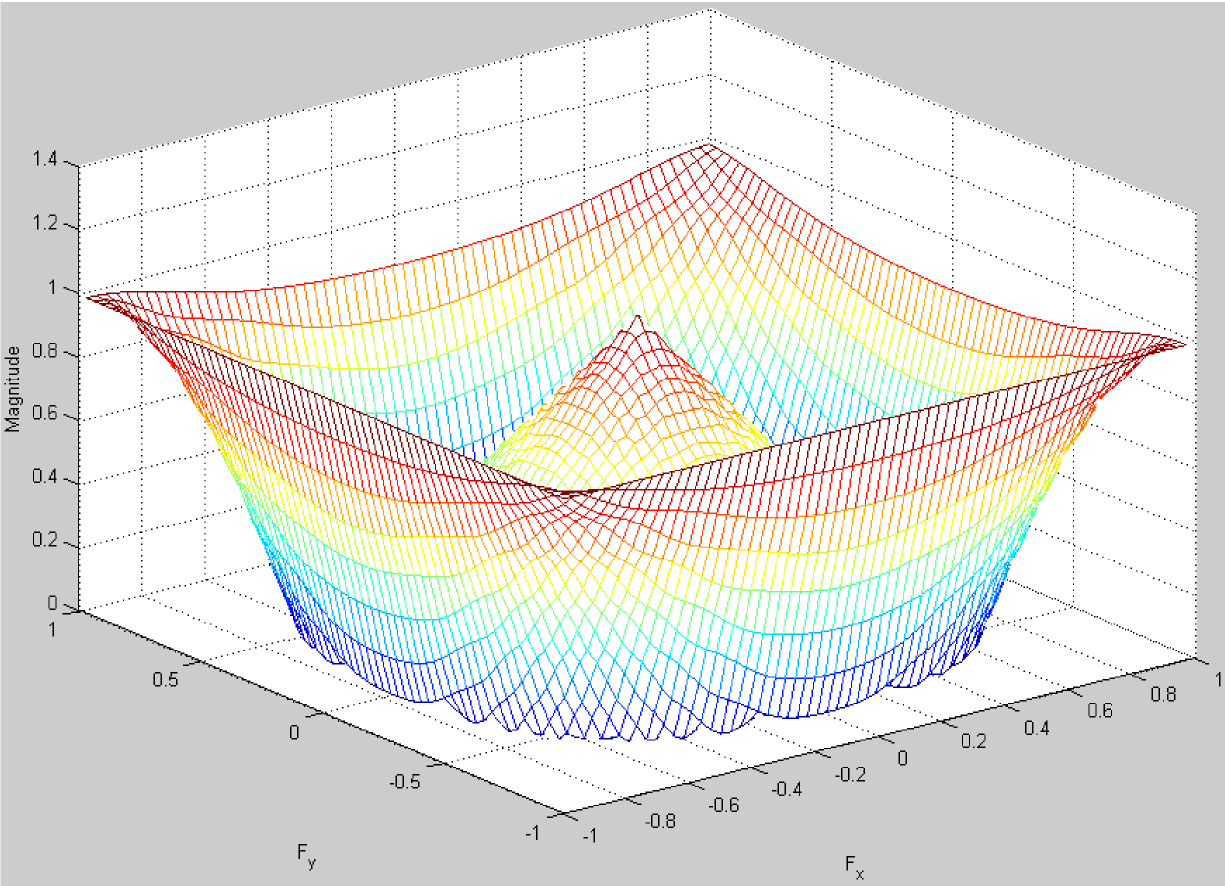
- Most energy at most frequent grey values
- Pairwise pulses mirrored at the Nyquist frequency
- Normed frequency to be attenuated:
- $f_0 = 493 / 630 \approx 0.7825$

The Magnitude Response of the Filter (2D View)



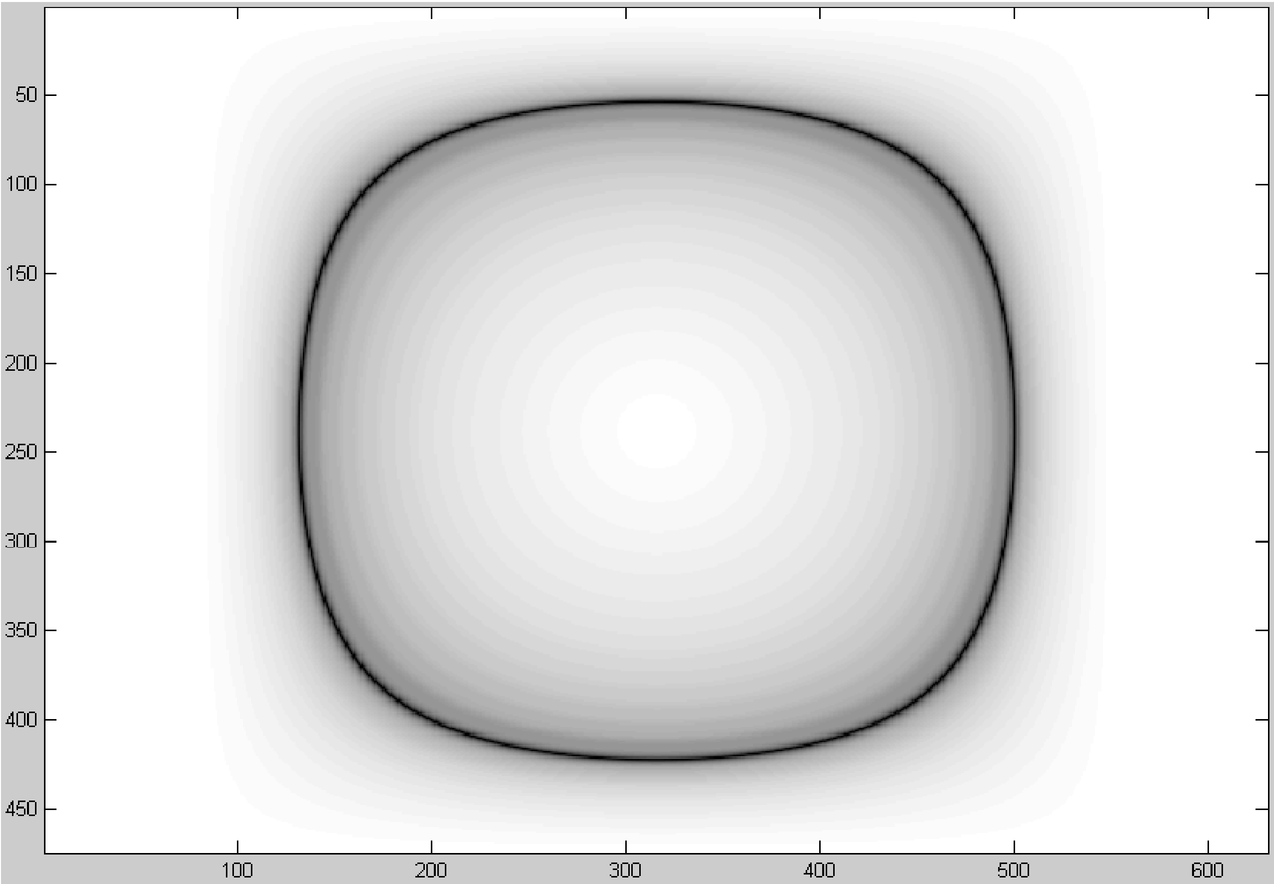
- Band reject filter (here: notch filter)
- Attenuates image at the normalised frequency $f_0 = 0.7825$

The Magnitude Response of the Filter (3D View)



- Band reject filter (here: notch filter)
- Attenuates image at the normalised frequency $f_0 = 0.7825$

Transfer Function H(x,y) (= the Filter Function's Spectrum)



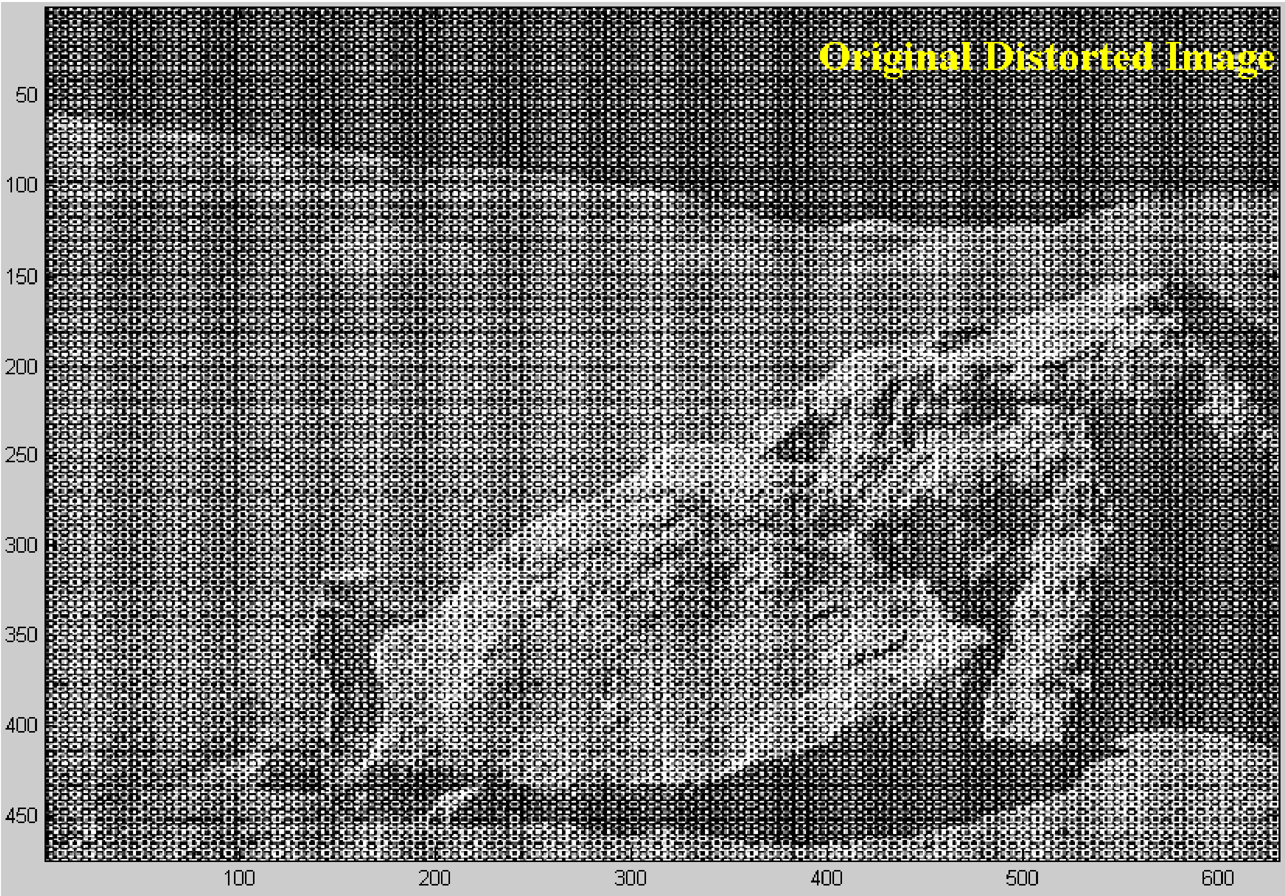
- Based on the convolution theorem

- $g(x,y) = f(x,y) * h(x,y)$

- $G(x,y) = F(x,y) \cdot H(x,y)$

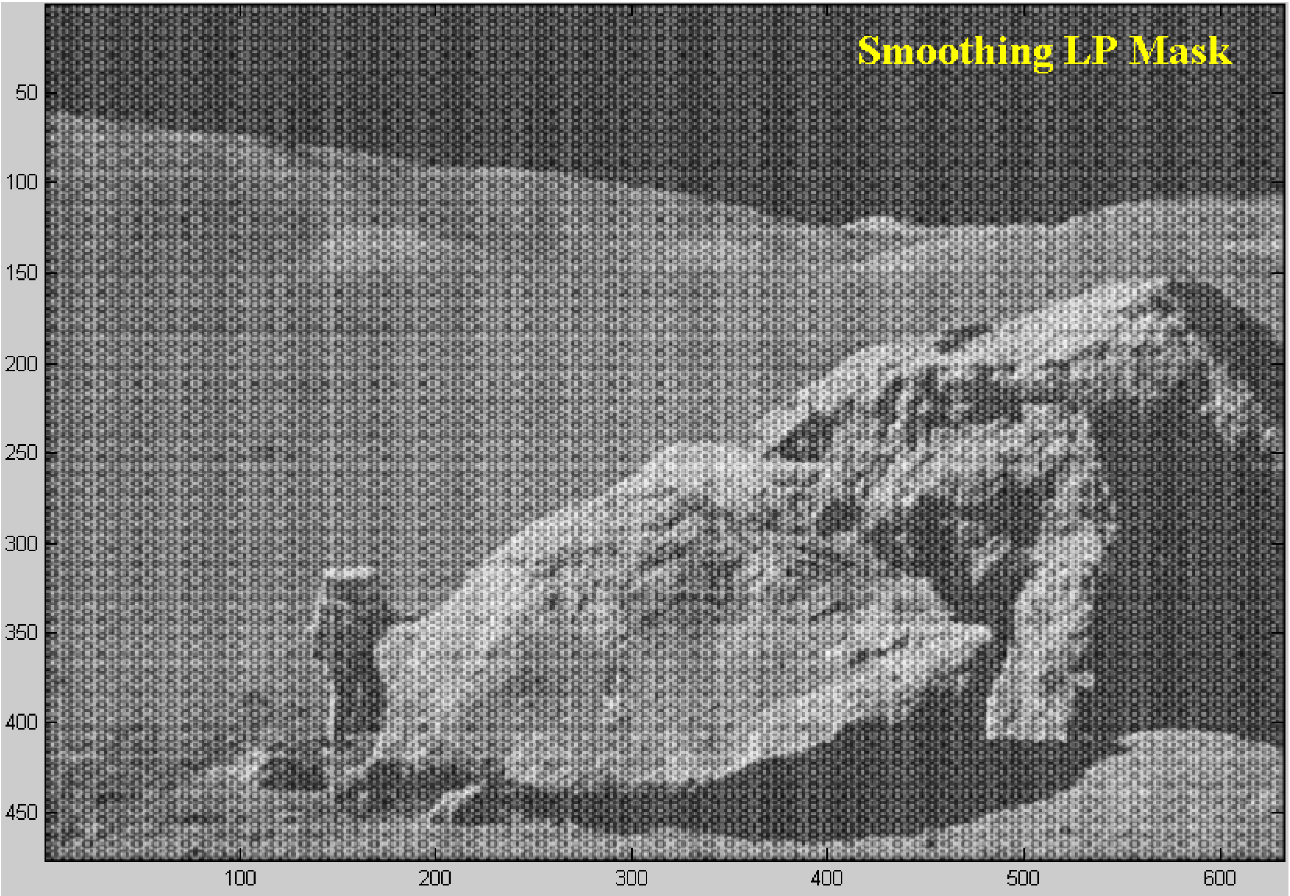
- Simply multiply transfer function with the image's spectrum

I) Result: Original Image f(x,y)



- Attenuation of the periodic disturbing signal by using a simple band reject filter

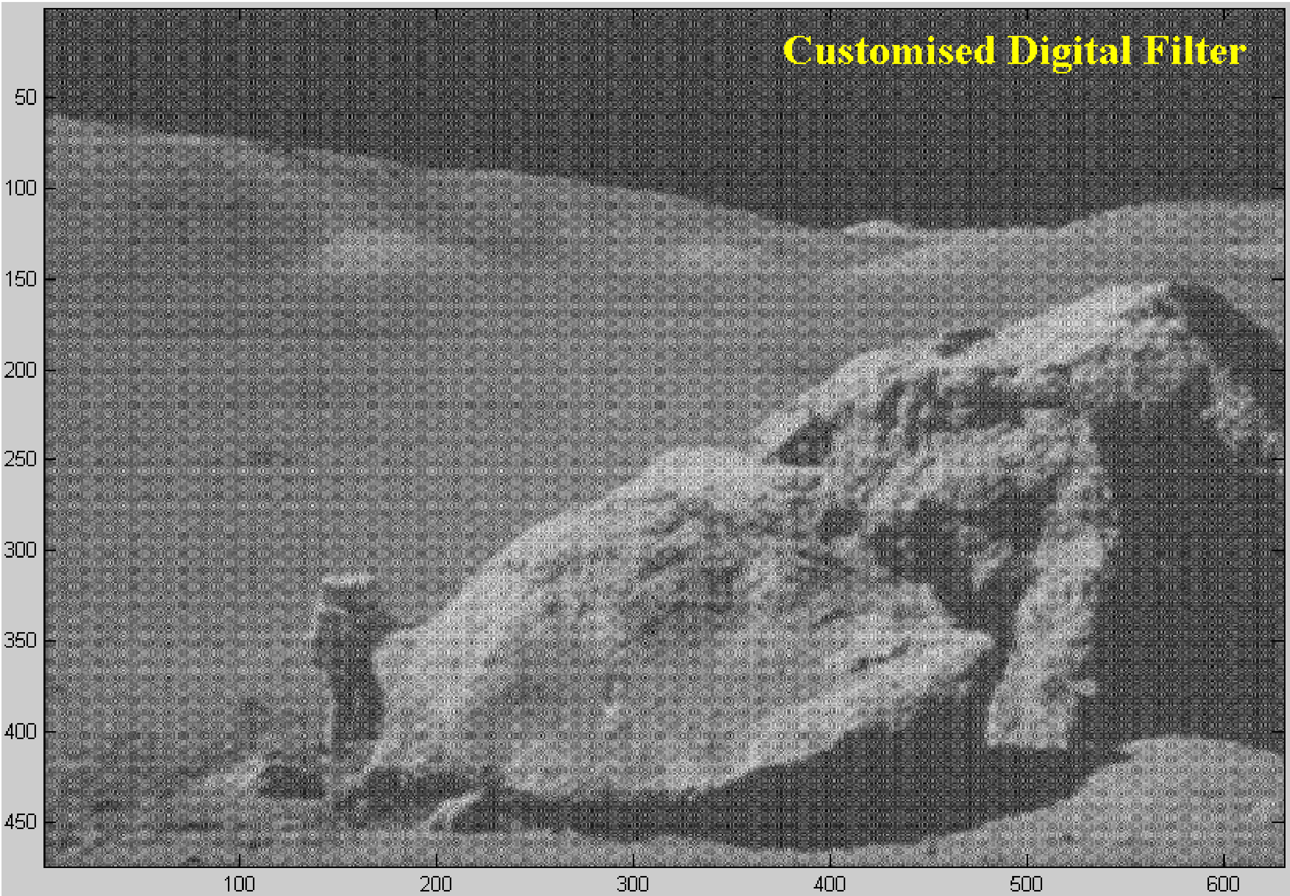
II) Result: Spatial Filtered Image g(x,y) using the Mask



$$mask = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

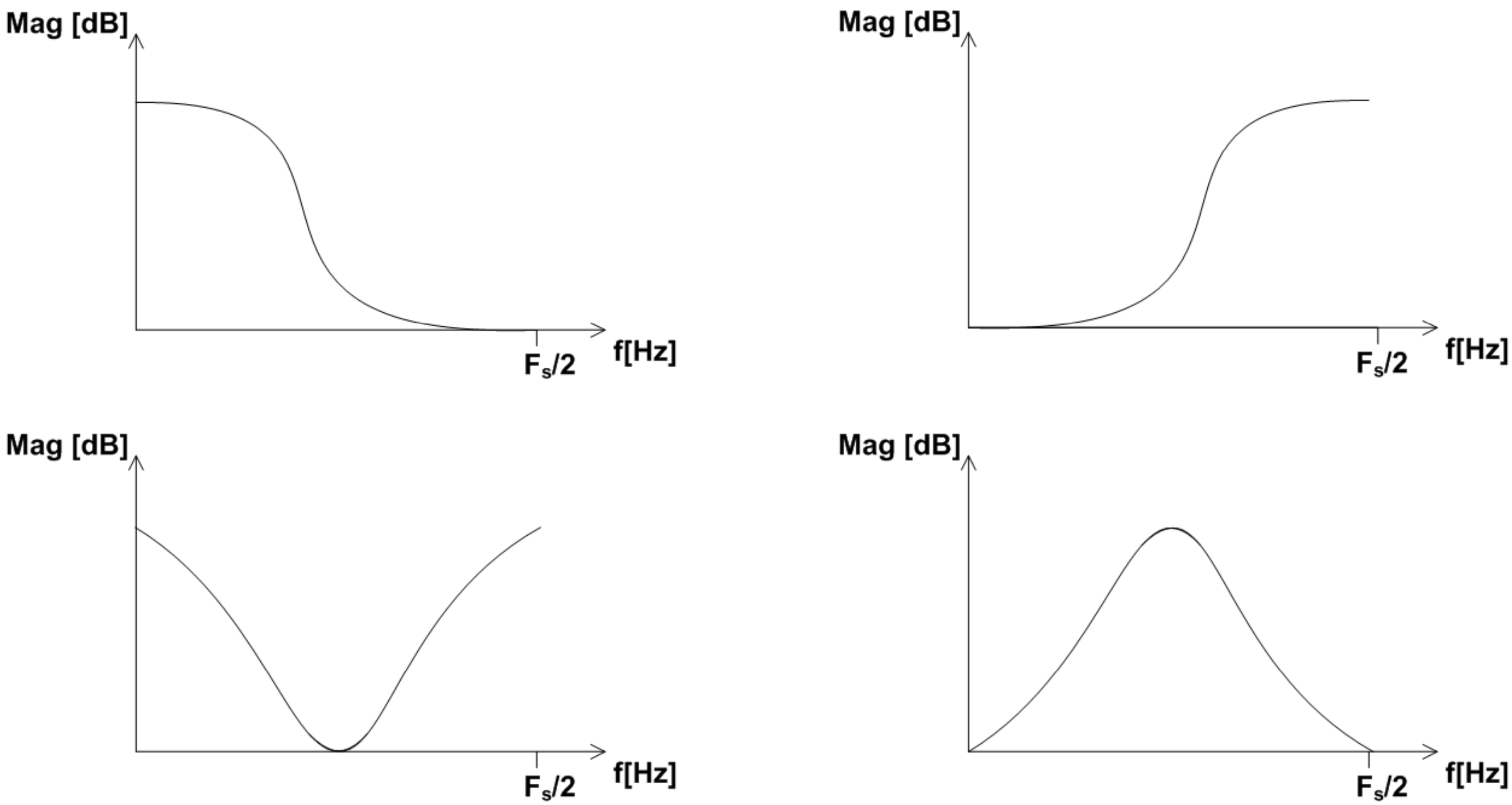
- Attenuation of the periodic disturbing signal by using a simple band reject filter

III) Result: Filtered Image g(x,y) using the Digital Filter

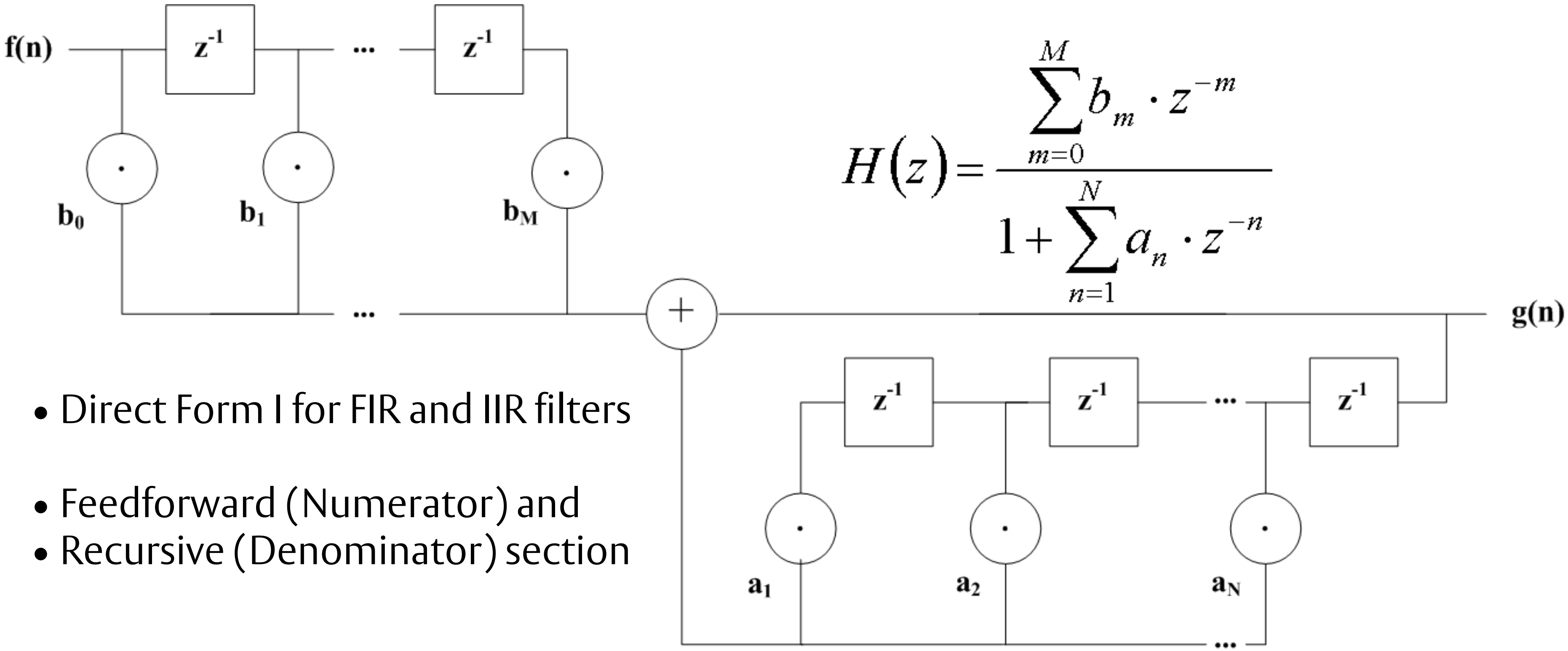


- Attenuation of the periodic disturbing signal by using a simple band reject filter

Filter Theory and Design – Basic Filter Types



Filter Theory and Design – Digital Filter Structure

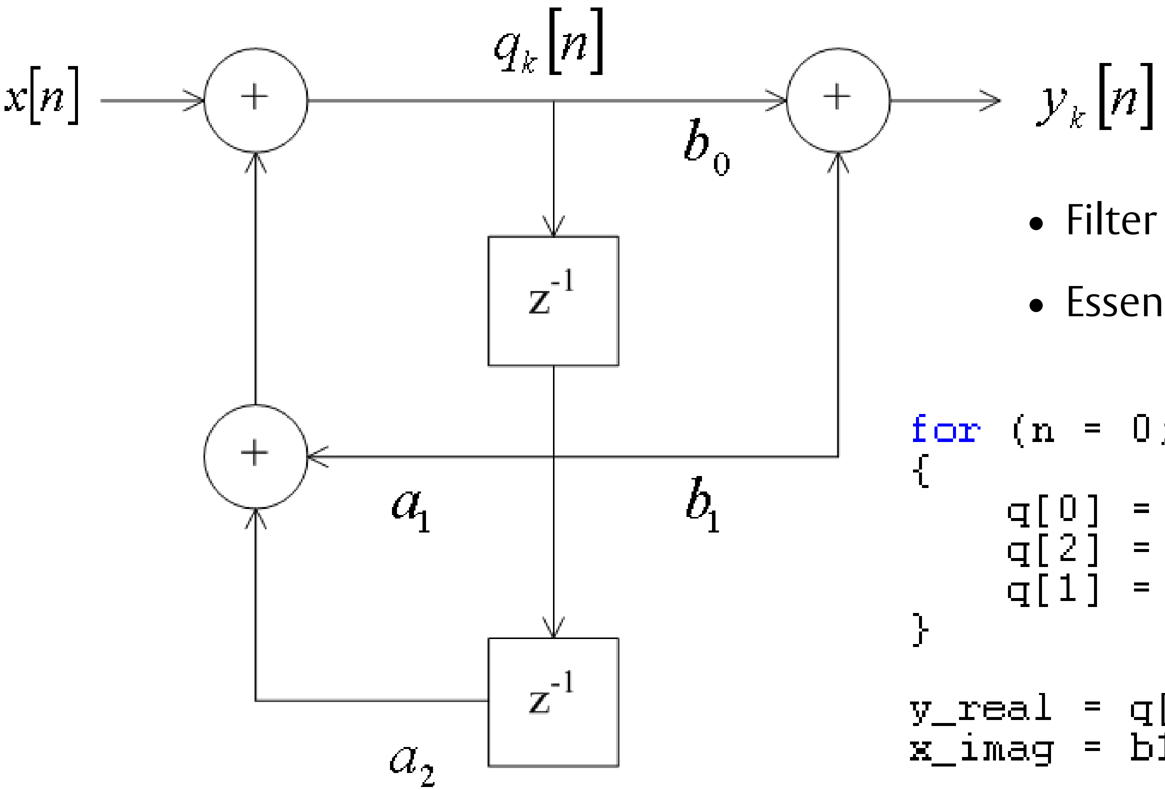


$$H(z) = \frac{\sum_{m=0}^{M} b_m \cdot z^{-m}}{1 + \sum_{n=1}^{N} a_n \cdot z^{-n}}$$

- Direct Form I for FIR and IIR filters

- Feedforward (Numerator) and
- Recursive (Denominator) section

Filter Theory and Design – Digital Filters in Practice

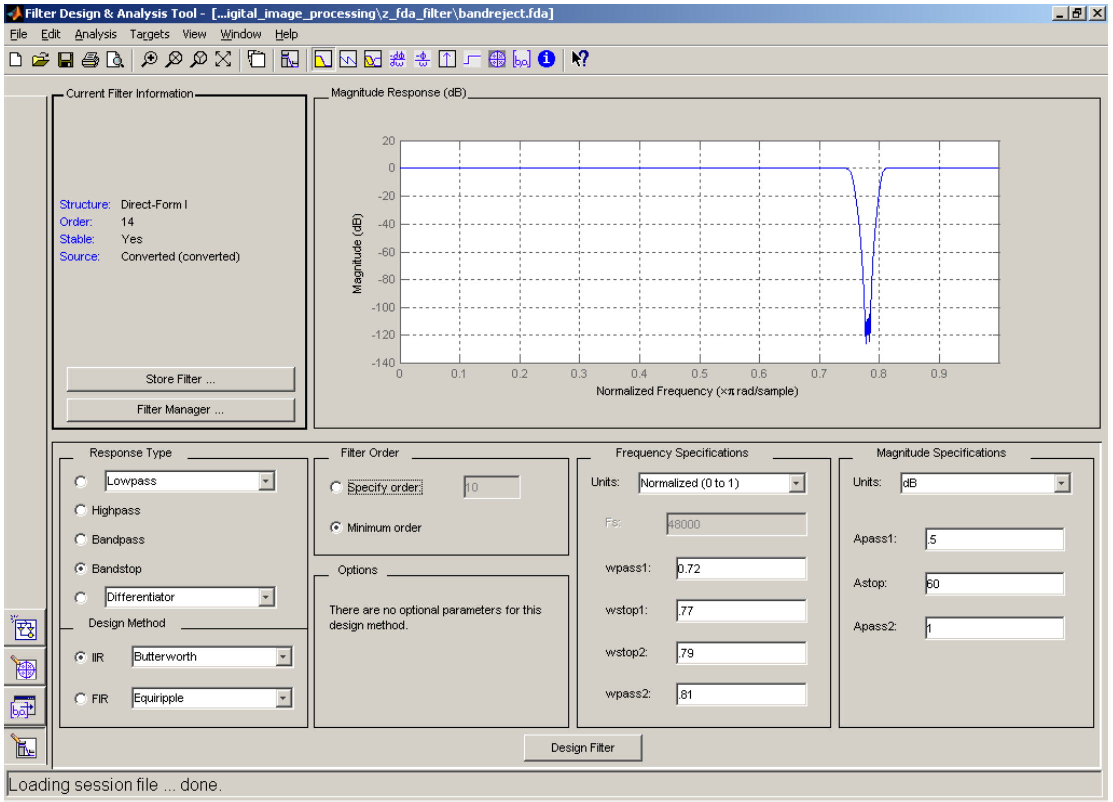


- Filter structure easy to implement
- Essential: Filter Coefficients

```
for (n = 0; n < length; n++)
{
    q[0] = a1 * q[1] - q[2] + x[n];
    q[2] = q[1];
    q[1] = q[0];
}

y_real = q[0] - b1_real * q[1];
x_imag = b1_imag * q[1];
```

Filter Theory and Design – The Filter Coefficients



- Pole-Zero-Diagram
- Tools from the WWW

- MatLab: >> fdatool
- Normed frequency to be attenuated: $f_0 \approx 0.78$

Case Study Conclusions

- Image enhancement is a necessary pre-processing step before segmentation
  - There are simple spatial masks for a rough enhancement
  - The use of customised digital filters is a more accurate alternative

- There are two steps in the design and implementation of Digital Filters
  - Fix the structure of the (FIR or IIR) filter in code
  - Compute the filter coefficient with a Design & Analysis tool (e.g. MatLab)

- However
  - Frequency domain techniques alone are insufficient
  - Take spatial domain and Wavelets into account

Overall Summary

- Overview provided of common frequency enhancement techniques
  - Introduction to Fourier Analysis
  - Application to Enhancement
  - Case Study