IIB 4F8: Image Processing and Image Coding

Handout 0: Introduction

J Lasenby

Signal Processing and Communications Group,
Engineering Department,
Cambridge, UK

Lent 2016

Introduction: 2d Image Processing

An **image** can take many forms and is not restricted to visual images.

Examples

1. Luminance of objects in a scene (ordinary camera)
2. Absorption characteristics of body tissue (x-ray camera)
3. Electron production of specimen scanned by electron beam (scanning electron microscope)
4. Temperature profile of a region (infra-red imaging)
5. Sidescan sonar (acoustic reflectance)
6. Seismic surveying, Earthquakes and seismic charges (often 3 or even 4 dimensional)
7. Radio-telescope images

Some examples

Some examples of the above are shown below:

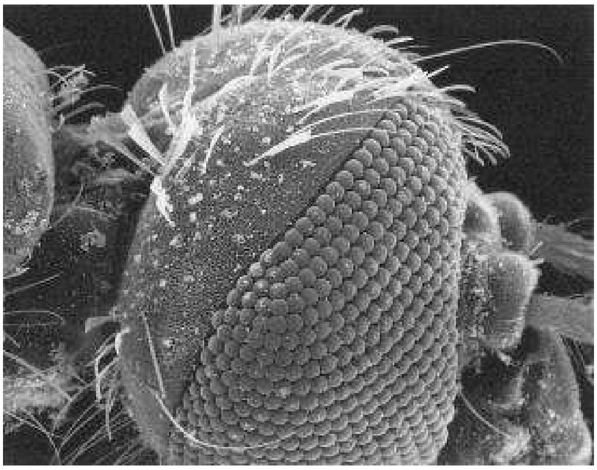


Figure 1: Scanning electron microscope image of a mosquito's head illustrating the compound eye structure (x200)

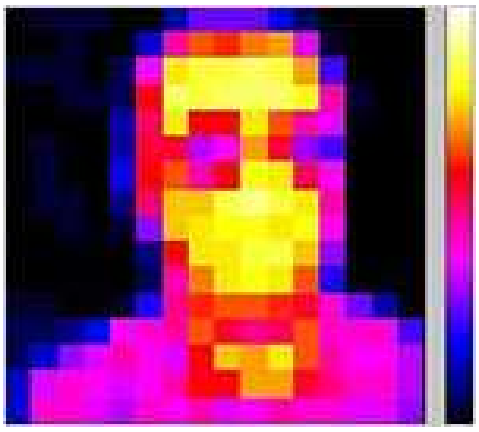


Figure 2: Image of a head and shoulders taken by a 16×16 thermal (IR) imaging array

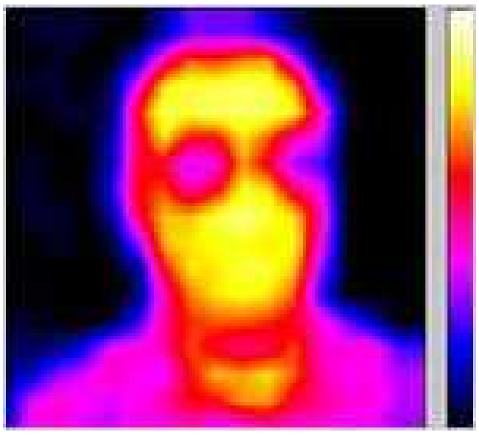


Figure 3: The 16×16 image interpolated to 128×128

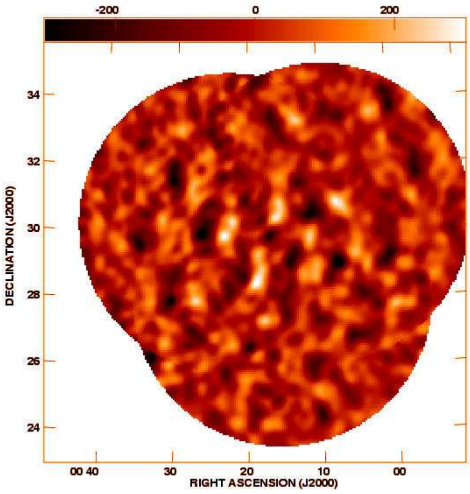


Figure 4: False colour image of the cosmic microwave background (CMB) radiation measured with the Very Small Array (VSA), Cambridge (2002)

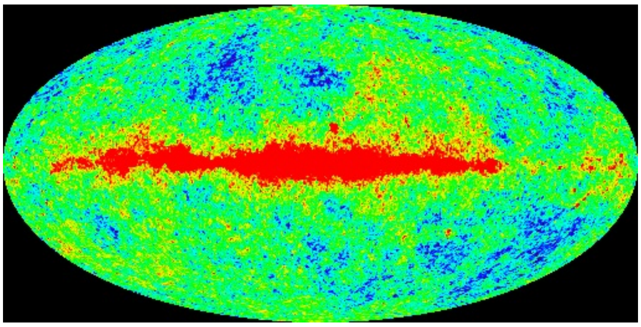


Figure 5: Map of the CMB taken with the WMAP telescope (2003)

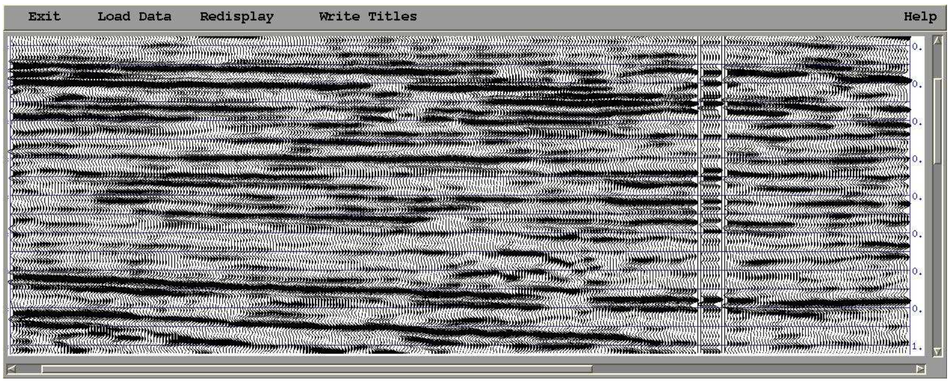


Figure 6: Stack of scans from a seismic experiment – each trace is the return over time at different distances from the initial source

Applications

Given the importance of visual images to humans, the range of applications is vast. The following list is a tiny subset of possible applications.

Astronomical imaging systems: In the past, cameras mounted on space-craft were limited in size, weight and power consumption so that image quality was often relatively poor - low signal to noise ratio, blurring, geometrical distortion.

Computationally intensive image processing was therefore routinely applied to the images, and many sophisticated techniques were thereby developed. Probably the best known ‘recent’ example of this was the Hubble space telescope (HST) – next figure shows an example of a picture of galaxy M87 in the early days of the HST (before optical corrections were carried out by the shuttle) before and after image processing and then the image after repair of the telescope.

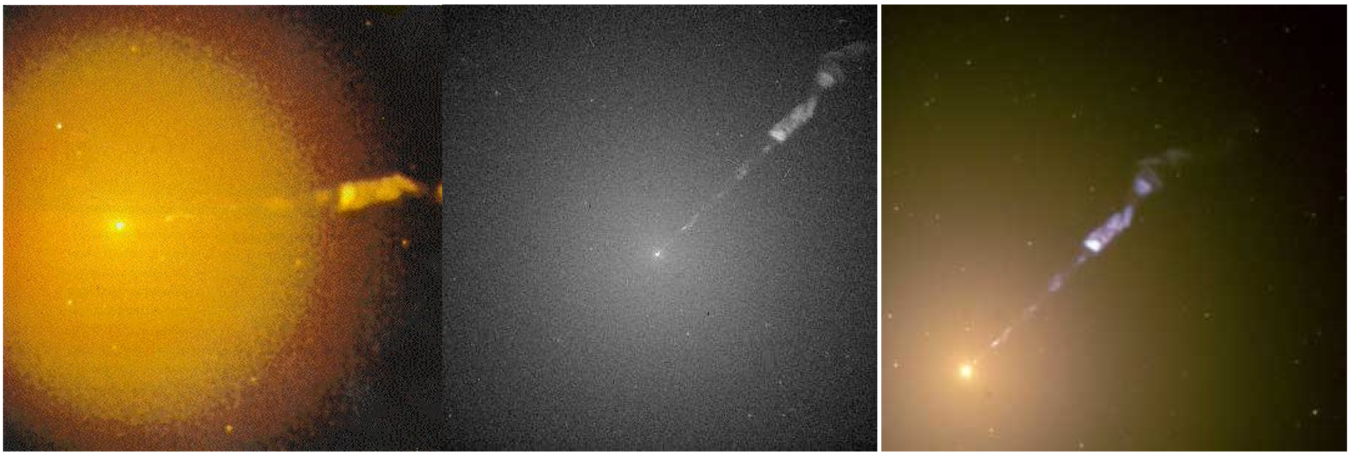


Figure 7: M87 as observed from the Hubble Space Telescope, before and after image processing and then repaired

Medical applications are of great importance. One of the earliest applications was enhancement of hairline fractures in x-ray images. Computed tomography (CT) is a technique of major importance in medicine and is based on the Projection-Slice Theorem which enables a 2-D image to be reconstructed from 1-D images. The image is reconstructed from the set of 1-D signals (projectors) taken at angles. The figure below shows the ideas behind this process

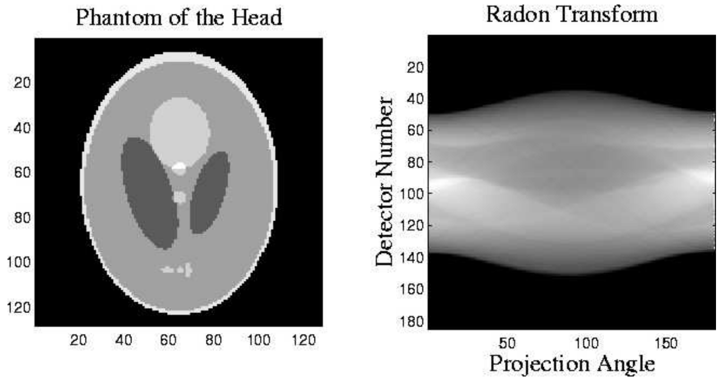


Figure 8: A simulated 2d image of the head and its radon transform – given a radon transform made up from 1d slices, the 2d image can be inferred. $R(L) = \int_L f(x) ds(x)$, L line in 2D, s arclength.

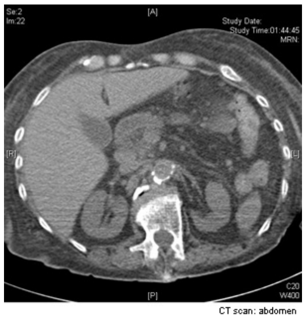


Figure 9: Abdominal CT scan

Forensic applications. Deblurring of photographs and still images, fingerprint identification, etc. In a past case of the abduction of a teenage girl, CCT footage of the last sighting of her underwent image processing to remove not *blur*, but *glare* which was present in the images. The difference glare can make is illustrated in the pictures below:

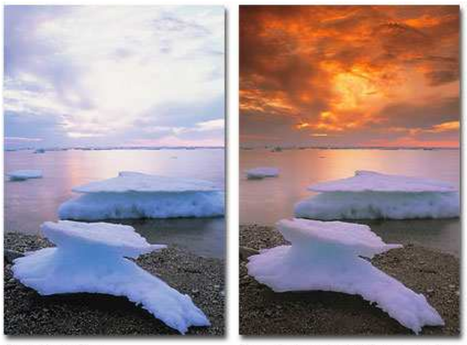


Photo 4A - No Filter

Photo 4B - 3-stop hard GND + warming filter

Figure 10: Left hand image is with glare, right hand image is without glare (in these pictures glare is actually removed with a polarizer)

Movies and Television: flicker (interpolated frames - need motion estimation), degradations etc. The figure below gives some idea of removal of degradations from video footage.

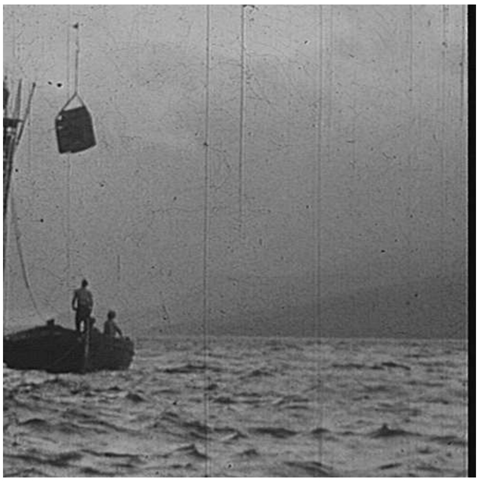


Figure 11: Example of degraded frame from video footage

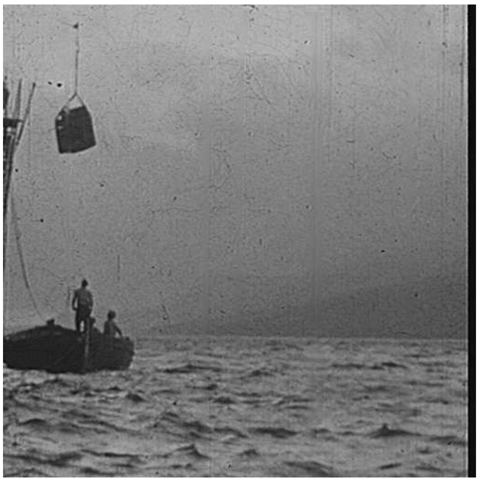


Figure 12: Restored frame, using blotch and line removal

Image coding for low bandwidth environments: we wish to compress our image data in as lossless a way as is possible in order to transmit the data efficiently (second half of course).

Computer vision – this deals with a sequence of images over time, or multiple sequences of images from which we estimate the full 3d reconstruction moving in time.

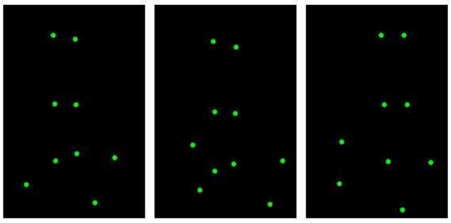


Figure 13: Three camera frames of a multiple camera setup observing a marked moving subject

Disparity Images from stereo cameras are used widely in computer vision to estimate depth and perform 3D reconstruction – following images show left and right images and the resulting disparity map.

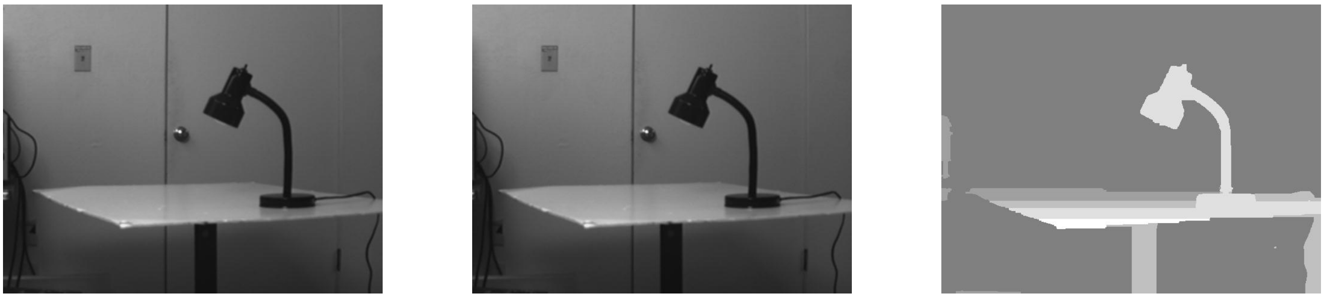


Figure 14: Left, Right and Disparity Images

[images taken from `http://vision.stanford.edu/`]

This is a very small sample of applications – as cheap computing becomes increasingly available, so image processing techniques will impinge on many more areas of everyday life, being available on mobile devices (this is already the case).

Image Representation and Modelling

Image processing can be broken down into a number of (overlapping) areas as shown below.

Perception Models

- visual perception of contrast, spatial frequencies and colour
- image fidelity models
- temporal/scene perception

Local Models

- Sampling and reconstruction
- image quantisation
- deterministic models
- statistical models

Global Models

- scene analysis/AI model
- sequential and clustering models
- image understanding models

We will concentrate on local models but some areas (image coding) must take account of results obtained from perception models. One can split our *local model* image processing up into several ‘areas’.

Image Enhancement

The objective of image enhancement is to accentuate particular features of an image in such a way that subsequent display is improved. The techniques do not introduce additional information. Examples are:

- Edge enhancement
- Grey level histogram equalisation
- Pseudo-colour (eg Matlab)

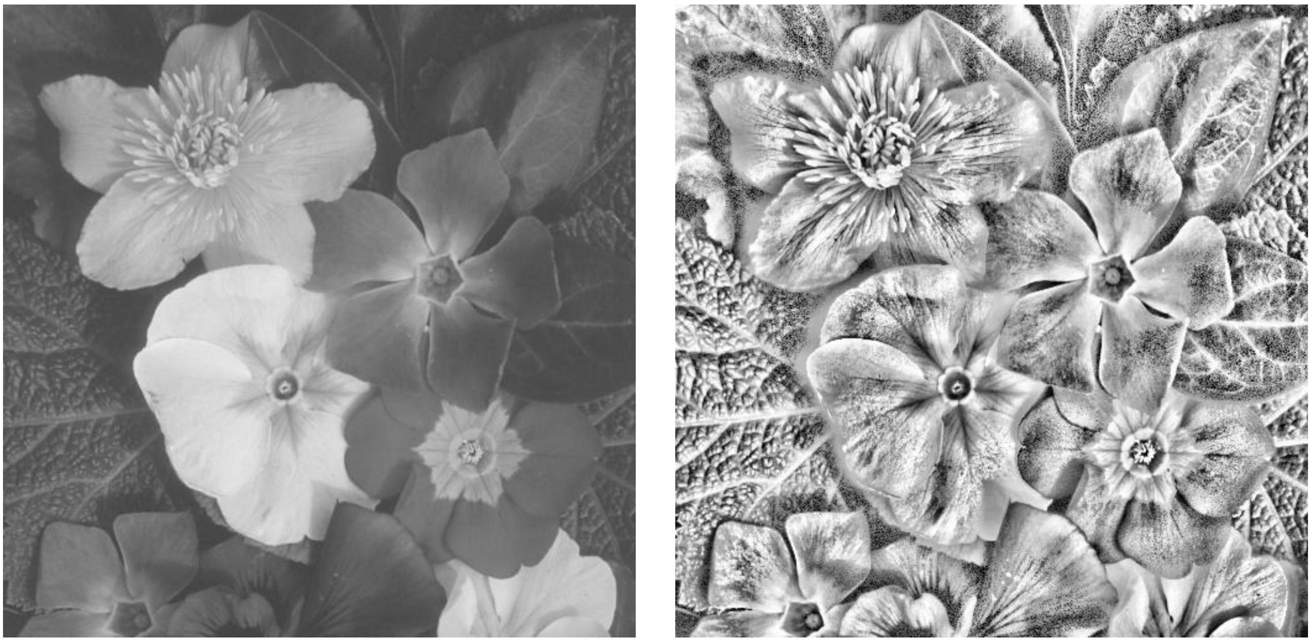


Figure 15: Original image (left) and application of adaptive histogram equalisation/contrast enhancement (right)

See `http://alex.mdag.org/improc`.

Image Restoration

The objective is to restore a degraded image to its original form. An observed image can often be modelled as:

$$f_{obs}(u_1, u_2) = \iint h(u_1 - u_1', u_2 - u_2') f_{true}(u_1', u_2')\; du_1'\; du_2' + n(u_1, u_2)$$

where the integral is a convolution function and h is the impulse response or point spread function of the imaging system and n is additive noise. The objective of image restoration in this case would be to estimate the original image f_{true} from the observed degraded image f_{obs}.

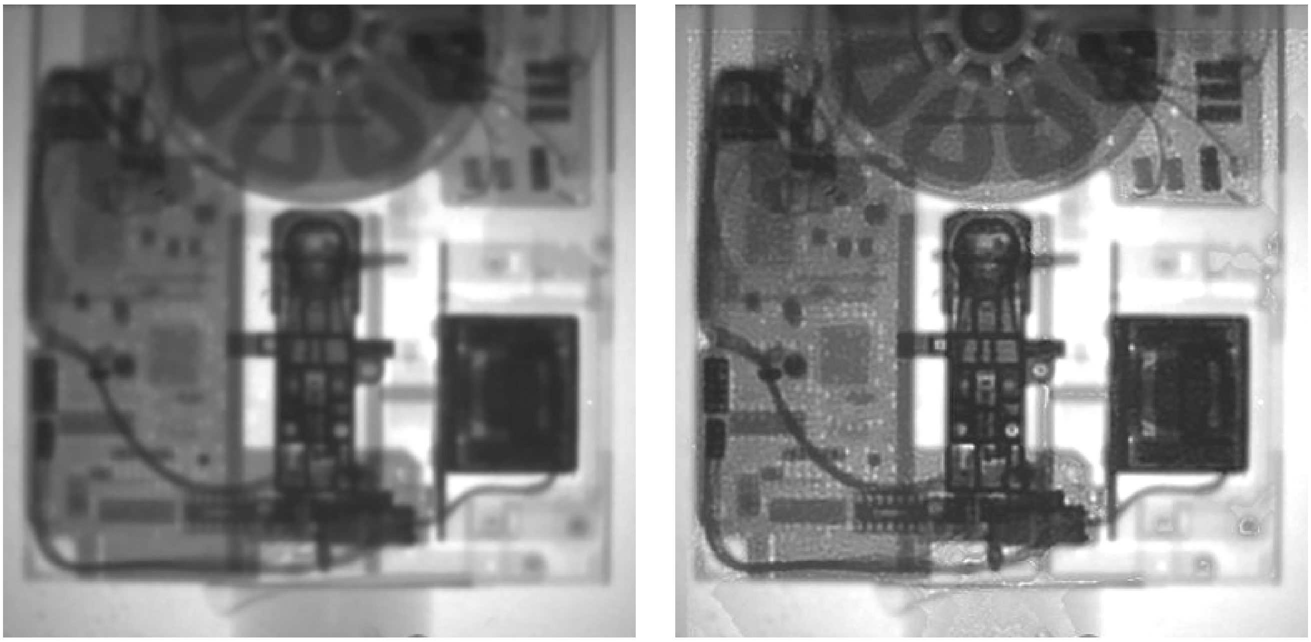


Figure 16: Original image (left) and deconvolved image (right)

See `http://www.mirametrics.com/brief_maxent_lab.htm`.

Image Analysis

Image analysis is concerned with extracting features from an image for subsequent analysis, or processing. Reading car number plates, inspection of items on conveyor belts etc.

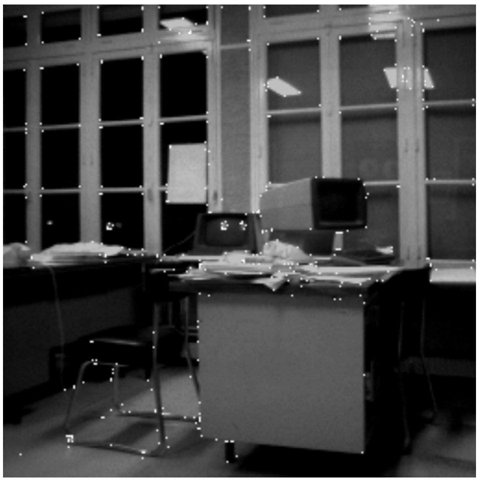


Figure 17: Extraction of corners from an indoor scene

Another example of using feature extraction for subsequent inference.

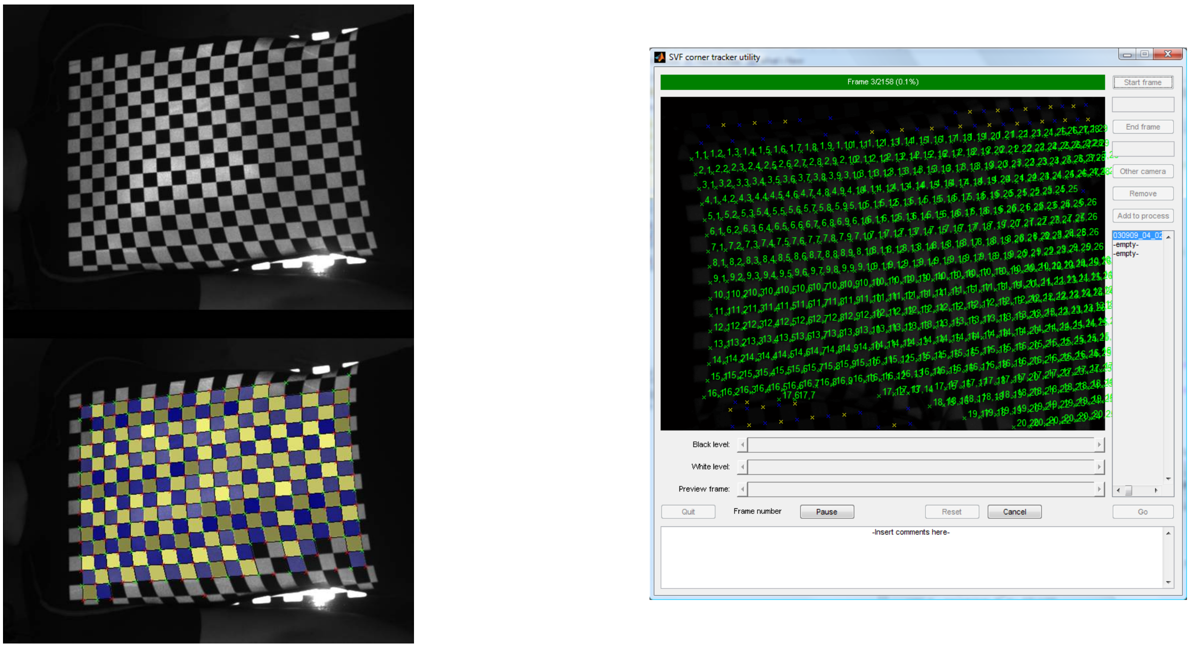


Figure 18: Dynamic surface reconstruction from grid point tracking

Course Contents and books

- 2d continuous Fourier transform and its properties
- Linear spatially invariant systems (LSI)
- Image digitisation and sampling – sampling schemes
- The 2d discrete Fourier transform (and mention 2d z -transform)
- Ideal image filters – the importance of phase
- Digital filter design (2d)
- Deconvolution of images
- Image enhancement
- 1. Fundamentals of digital image processing: Anil Jain (Prentice Hall) 2. Digital image processing: Gonzalez and Woods (Prentice Hall) 3. Two-dimensional signal and image processing: J S Lim (Prentice Hall)

J. Lasenby (2016)

IIB 4F8: Image Processing and Image Coding

Handout 1: 2D Fourier Transforms and Linear Systems

J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

Course notes, Matlab programs and Example Sheet available for download from
www-sigproc.eng.cam.ac.uk/~jl

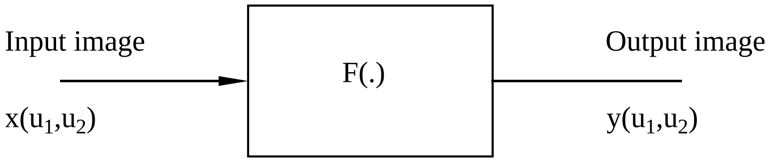
Now also available on the CUED Moodle site:
https://www.vle.cam.ac.uk/

IB Signal and Data Analysis notes available from
www-sigproc.eng.cam.ac.uk/~jl

... and from Prof Simon Godsill's website:
www-sigproc.eng.cam.ac.uk/~sjg

Multidimensional Signals and Systems

Will discuss 2-D signals and systems (but readily generalised to higher dimensions). Consider the system below which has input $x(u_1, u_2)$ and output $y(u_1, u_2)$



Input image

$x(u_1,u_2)$

F(.)

Output image

$y(u_1,u_2)$

$$y(u_1, u_2) = F[x(u_1, u_2)]$$

Want to characterise the functional mapping $F[\cdot]$ between input and output in terms of concepts such as impulse response and frequency response. First need to generalise the concept of the Fourier transform to multiple dimensions.

The 2d Fourier Transform

The 1-D Fourier transform $G(\omega)$ of a function $g(t)$ is given by:

$$G(\omega) = \int_{-\infty}^{\infty} g(t)\, e^{-j\omega t}\, dt$$

$$g(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega)\, e^{j\omega t}\, d\omega$$

The 2-D Fourier transform of a spatial function $g(u_1, u_2)$ is given by:

$$\boxed{G(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2}$$

..with inverse 2DFT given by

$$\boxed{g(u_1, u_2) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\omega_1, \omega_2)\, e^{j(\omega_1 u_1 + \omega_2 u_2)}\, d\omega_1\, d\omega_2}$$

Proving Inverse 2d Fourier Transform

Do this by direct substitution:

$$\begin{aligned}
& FT(g(u_1,u_2)) = \\
= \quad & \frac{1}{(2\pi)^2}\int\int\left[\int\int G(\omega_1',\omega_2')\, e^{j(\omega_1' u_1+\omega_2' u_2)}\, d\omega_1'\, d\omega_2'\right] e^{-j(\omega_1 u_1+\omega_2 u_2)}\, du_1\, du_2 \\
= \quad & \frac{1}{(2\pi)^2}\int\int G(\omega_1',\omega_2')\left[\int\int e^{j((\omega_1'-\omega_1)u_1+(\omega_2'-\omega_2)u_2)}\, du_1\, du_2\right] d\omega_1'\, d\omega_2' \\
= \quad & \frac{1}{(2\pi)^2}\int\int (2\pi)^2 G(\omega_1',\omega_2')\,\delta(\omega_1'-\omega_1)\delta(\omega_2'-\omega_2) d\omega_1'\, d\omega_2' \\
= \quad & G(\omega_1,\omega_2)
\end{aligned}$$

Where we have used the result that $\int_{-\infty}^{\infty}\exp(\pm j\omega t)d\omega = 2\pi\delta(t)$ – see later.

Separability of 2d Fourier Transform

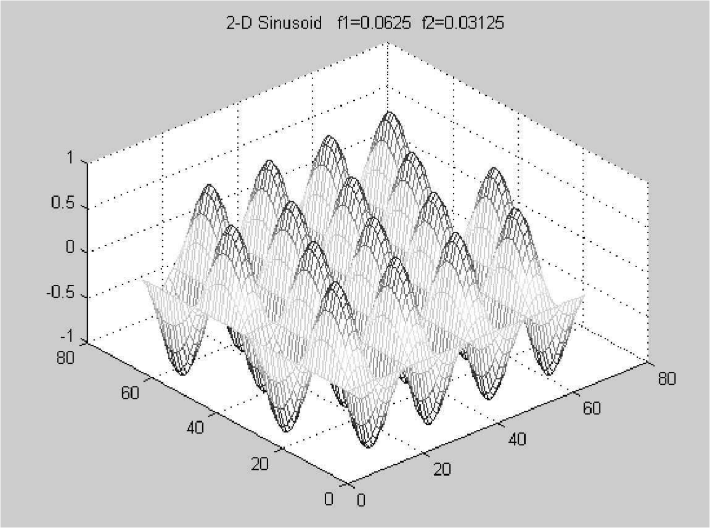
Note that the 2d Fourier transform can be rewritten as:

$$G(\omega_1,\omega_2) = \int_{-\infty}^{\infty} e^{-j\omega_1 u_1}\left\{\int_{-\infty}^{\infty} g(u_1,u_2)\, e^{-j\omega_2 u_2}\, du_2\right\} du_1$$

Inner integral is simply the FT of the waveform corresponding to moving along a line in the u_2-direction at some fixed position u_1. This gives a function of 'vertical' position u_1 and 'horizontal' spatial frequency ω_2. Outer integral is the Fourier transform of this function in the u_1 direction.

Separability of 2d Fourier Transform – illustration

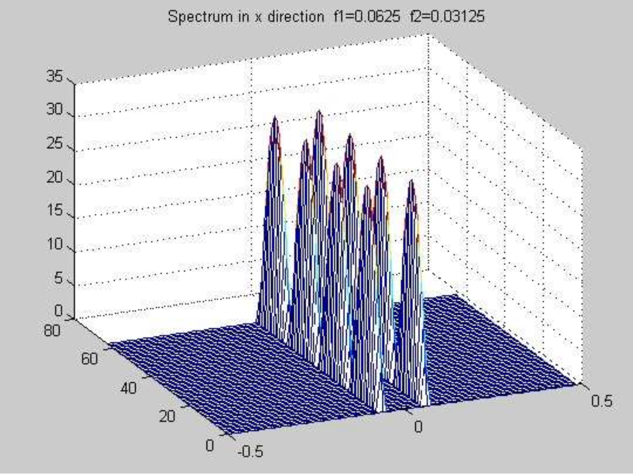
This may be illustrated by considering the 2-dimensional sinewave $g(x,y) = \sin\Omega_1 x\ \sin\Omega_2 y$, which is shown in the figure for $\Omega_1 = 2\pi f_1,\ f_1 = \frac{1}{16}$Hz and $\Omega_2 = 2\pi f_2,\ f_2 = \frac{1}{32}$Hz.



2d sinewave example

If we first do the integration wrt y we get

$$\begin{aligned} F(x,\omega_2) &= \sin\Omega_1 x \left\{ \int_{-\infty}^{\infty} \sin\Omega_2 y\ e^{-j\omega_2 y}\, dy \right\} \\ &= j\pi \sin\Omega_1 x\ \ [\delta(\omega_2+\Omega_2) - \delta(\omega_2-\Omega_2)] \end{aligned}$$



2d sinewave example cont...

If we now do the integration in the x direction we obtain

$$G(\omega_1, \omega_2) = j\pi \left[\delta(\omega_2 + \Omega_2) - \delta(\omega_2 - \Omega_2)\right] \int_{-\infty}^{\infty} \sin \Omega_1 x \; e^{-j\omega_1 x} \, dx$$

$$= -(\pi)^2 \left[\delta(\omega_1 + \Omega_1) - \delta(\omega_1 - \Omega_1)\right] \left[\delta(\omega_2 + \Omega_2) - \delta(\omega_2 - \Omega_2)\right]$$

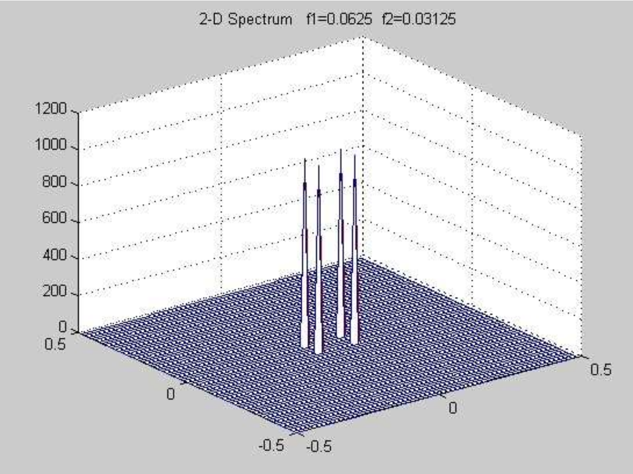


Illustration of spatial frequencies in images

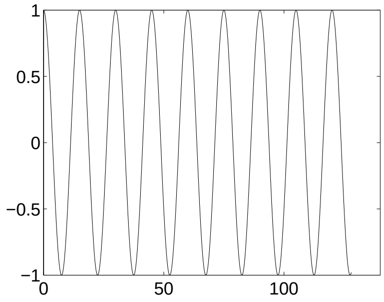


Figure 1: 1d sine wave

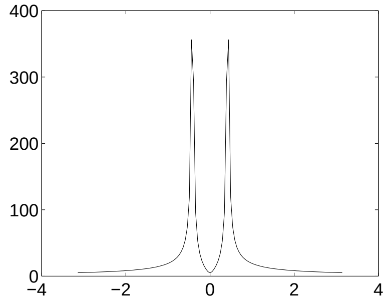


Figure 2: Spectrum of the 1d sine wave

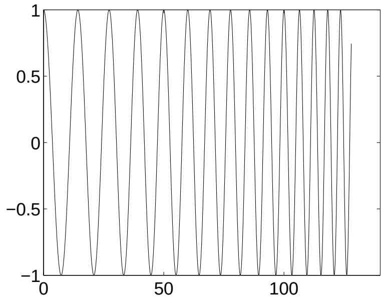


Figure 3: 1d sine wave – varying frequency

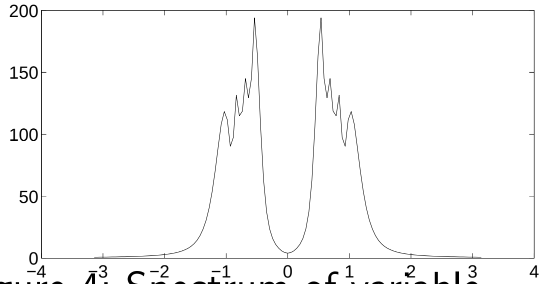


Figure 4: Spectrum of variable frequency sine wave

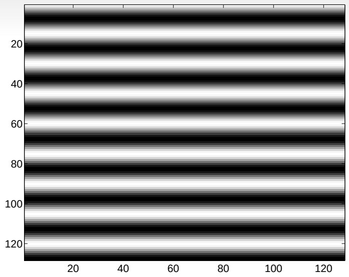


Figure 5: 2d sine wave

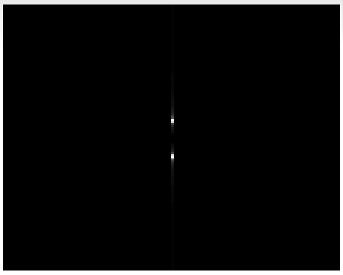


Figure 6: Spectrum of the 2d sine wave

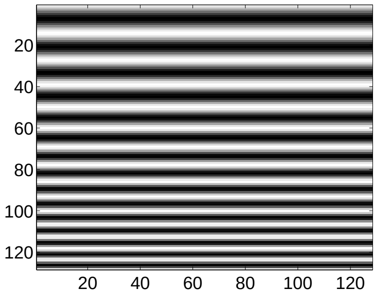


Figure 7: 2d sine wave – varying frequency [spatial tilt]

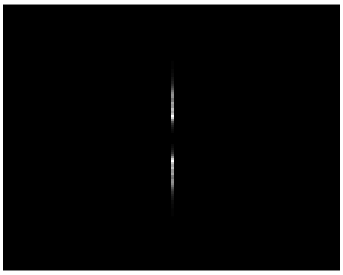


Figure 8: Spectrum of variable frequency 2d sine wave

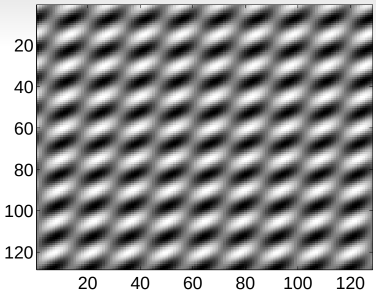


Figure 9: 2d texture

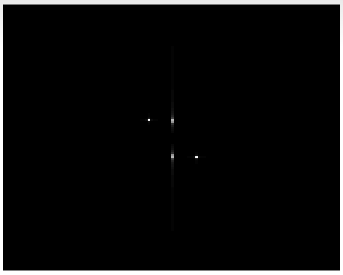


Figure 10: Spectrum of the texture

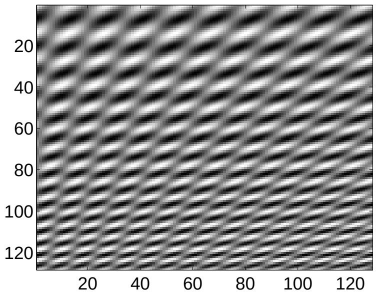


Figure 11: 2d texture – varying frequency [spatial tilt]

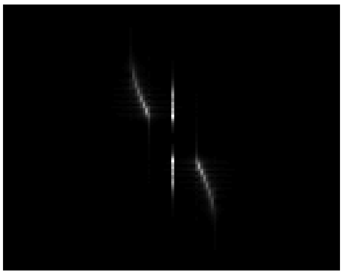


Figure 12: Spectrum of variable frequency 2d texture

Example of using texture to extract shape

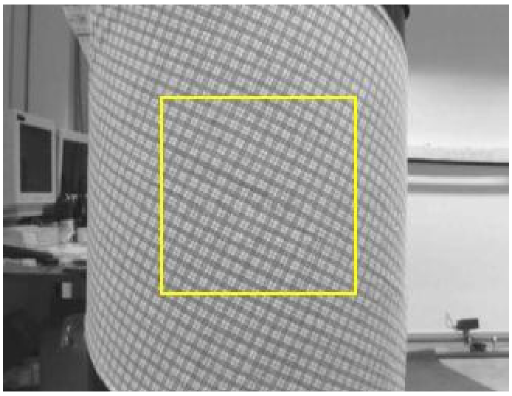


Figure 13: Texture laid onto cylindrical pipe

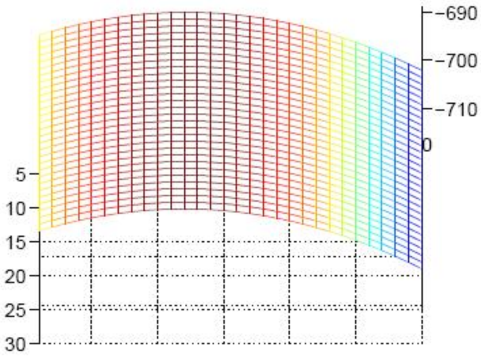


Figure 14: Shape from extraction of frequencies from the texture

(Fabio Galasso, PhD Thesis, University of Cambridge 2009).

Properties of 2d Fourier Transforms: Spatial Shift

Consider a spatially shifted version $g(u_1 - \mu_1, u_2 - \mu_2)$ of the image $g(u_1, u_2)$. The FT of the shifted image is:

$$G^{'}(\omega_1, \omega_2) = \iint_{-\infty}^{\infty} g(u_1 - \mu_1, u_2 - \mu_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

Let $u_1 - \mu_1 = u_1^{'}$ and $u_2 - \mu_2 = u_2^{'}$

$$\begin{aligned} G^{'}(\omega_1, \omega_2) &= \iint_{-\infty}^{\infty} g(u_1^{'}, u_2^{'})\, e^{-j[(\omega_1(u_1^{'}+\mu_1)+\omega_2(u_2^{'}+\mu_2)]}\, du_1^{'}\, du_2^{'} \\ &= e^{-j(\mu_1\omega_1+\mu_2\omega_2)} \iint g(u_1^{'}, u_2^{'})\, e^{-j(\omega_1 u_1^{'}+\omega_2 u_2^{'})}\, du_1^{'}\, du_2^{'} \\ &= e^{-j(\mu_1\omega_1+\mu_2\omega_2)}\, G(\omega_1, \omega_2) \end{aligned}$$

$$\therefore \quad \boxed{g(u_1 - \mu_1, u_2 - \mu_2) \Leftrightarrow e^{-j(\mu_1\omega_1+\mu_2\omega_2)}\, G(\omega_1, \omega_2)}$$

Properties: Frequency Shift or Spatial Modulation

Consider a spatial-frequency shifted image transform $G(\omega_1 - \Omega_1, \omega_2 - \Omega_2)$. The FT is

$$G(\omega_1, \omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

replace ω_1 and ω_2 by $\omega_1 - \Omega_1$ and $\omega_2 - \Omega_2$

$$G(\omega_1 - \Omega_1, \omega_2 - \Omega_2) = \iint_{-\infty}^{\infty} g(u_1, u_2)\, e^{-j((\omega_1 - \Omega_1)u_1 + (\omega_2 - \Omega_2)u_2)}\, du_1\, du_2$$

$$= \iint_{-\infty}^{\infty} e^{j(\Omega_1 u_1 + \Omega_2 u_2)}\, g(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

$$\therefore \quad \boxed{g(u_1, u_2) e^{j(\Omega_1 u_1 + \Omega_2 u_2)} \Leftrightarrow G(\omega_1 - \Omega_1, \omega_2 - \Omega_2)}$$

Again, exactly as in the 1d case. In both cases, a shift in one domain causes multiplication by an exponential in the other domain.

Useful Fourier Transforms: Impulse function

The function $\delta(u_1, u_2)$ is an impulse occuring at the origin $u_1 = 0, u_2 = 0$ and the function $\delta(u_1 - \mu_1, u_2 - \mu_2)$ is an impulse occuring at $u_1 = \mu_1, u_2 = \mu_2$.
The impulse function is defined by:

$$\lim_{\epsilon \to 0} \int_{-\epsilon}^{\epsilon}\int_{-\epsilon}^{\epsilon} \delta(u_1, u_2)\, du_1\, du_2 = 1 \qquad \text{Unit area}$$

$$\iint_{-\infty}^{\infty} \delta(u_1 - \mu_1, u_2 - \mu_2)\, f(u_1, u_2)\, du_1\, du_2 = f(\mu_1, \mu_2) \qquad \text{Sifting property}$$

The Fourier transform of the 2-D impulse function is:

$$\Delta(\omega_1, \omega_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2 = 1$$

$$\therefore \quad \boxed{\delta(u_1, u_2) \Leftrightarrow 1}$$

Useful Fourier Transforms: Constant level

Let $g(u_1, u_2) = k$ then

$$G(\omega_1, \omega_2) = \iint_{-\infty}^{\infty} k\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

We saw in the IB course that we could write

$$\lim_{a \to \infty} \frac{\sin at}{t} \to \pi\delta(t)$$

Exercise: Use the above identity to show that

$$\lim_{A \to \infty} \int_{-A}^{A} \exp(\pm j\omega t) d\omega = 2\pi\delta(t)$$

We can now use this result:

$$\begin{aligned} G(\omega_1, \omega_2) &= \iint_{-\infty}^{\infty} k\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2 \\ &= k(2\pi\delta(\omega_1))(2\pi\delta(\omega_2)) = k(2\pi)^2\delta(\omega_1, \omega_2) \end{aligned}$$

Useful Fourier Transforms: Constant level cont...

Indeed, physical intuition $\Longrightarrow$ spectrum of a constant image would be zero at all spatial frequencies other than $\omega_1 = \omega_2 = 0$. We can check our result by taking the inverse transform.

$$\frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} k(2\pi)^2\, \delta(\omega_1, \omega_2)\, e^{j(\omega_1 u_1 + \omega_2 u_2)}\, d\omega_1\, d\omega_2$$

$$= k$$

$$\therefore \quad \boxed{k \Leftrightarrow k\,(2\pi)^2\,\delta(\omega_1, \omega_2)}$$

Useful Fourier Transforms: Complex Exponential

Let $g(u_1, u_2) = e^{j(\Omega_1 u_1 + \Omega_2 u_2)}$ then

$$\begin{aligned} G(\omega_1, \omega_2) &= \iint_{-\infty}^{\infty} e^{j(\Omega_1 u_1 + \Omega_2 u_2)} \, e^{-j(\omega_1 u_1 + \omega_2 u_2)} \, du_1 \, du_2 \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-j[(\omega_1 - \Omega_1) u_1 + (\omega_2 - \Omega_2) u_2]} \, du_1 \, du_2 \\ &= (2\pi)^2 \delta(\omega_1 - \Omega_1, \omega_2 - \Omega_2) \end{aligned}$$

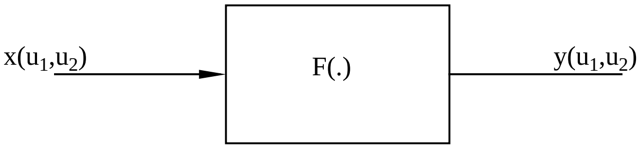
Note that it is also possible to see this result directly using the frequency shift theorem:

$$g(u_1, u_2) e^{j(\Omega_1 u_1 + \Omega_2 u_2)} \Leftrightarrow G(\omega_1 - \Omega_1, \omega_2 - \Omega_2)$$

In this case: $g(u_1, u_2) = 1$ and $G(\omega_1, \omega_2) = (2\pi)^2 \delta(\omega_1, \omega_2)$

$$\therefore \quad \boxed{e^{j(\Omega_1 u_1 + \Omega_2 u_2)} \Leftrightarrow (2\pi)^2 \delta(\omega_1 - \Omega_1, \omega_2 - \Omega_2)}$$

Linear Systems



$$y(u_1, u_2) = \mathcal{F}[x(u_1, u_2)]$$

Here $x(u_1, u_2)$ is the input to the linear system and $y(u_1, u_2)$ is the output. A system is linear if:

$$\mathcal{F}[a \, x_1(u_1, u_2) + b \, x_2(u_1, u_2)] = a \, \mathcal{F}[x_1(u_1, u_2)] + b \, \mathcal{F}[x_2(u_1, u_2)]$$

If the input to the system is a 2-d impulse function $\delta(u_1 - \mu_1, u_2 - \mu_2)$ positioned at $u_1 = \mu_1, u_2 = \mu_2$ then the resulting output image is termed the impulse response $h(u_1, u_2; \mu_1, \mu_2)$ of the system. That is:

$$h(u_1, u_2; \mu_1, \mu_2) = \mathcal{F}[\delta(u_1 - \mu_1, u_2 - \mu_2)]$$

Linear Systems: spatial invariance

If the system is *spatially invariant* then the shape of the impulse response is independent of the position of the impulse function – i.e. a translation of the input produces a translation of the output;

$$\mathcal{F}[\delta(u_1 - \mu_1, u_2 - \mu_2)] = h(u_1 - \mu_1, u_2 - \mu_2)$$

where:

$$h(u_1, u_2) = \mathcal{F}[\delta(u_1, u_2)]$$

The *region of support* R_h of the impulse response is the smallest closed region in the output plane outside of which the impulse response is zero. Finite impulse response (FIR) and infinite impulse response (IIR) systems have finite and infinite regions of support respectively.

When the 2-d signal is an image expressed in terms of greylevels it is always positive and the impulse response is often referred to as the *point spread function*

Linear Systems: convolution

For a Linear Time Invariant (LTI) 1-d system we can write the input $x(t)$ as an integral $x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$. If an impulse $\delta(t)$ produces an output $h(t)$, then by linearity, an input $x(t)$ will produce an output $y(t)$ given by

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\, h(t-\tau)\, d\tau$$

... the convolution integral (proper proof of this is more involved – see IA Maths notes). The corresponding results in 2d are

$$x(u_1, u_2) = \iint_{-\infty}^{\infty} x(u_1', u_2')\, \delta(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2'$$

$$y(u_1, u_2) = \iint_{-\infty}^{\infty} x(u_1', u_2')\, h(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2'$$

$$\text{or} \quad y(u_1, u_2) = \iint_{-\infty}^{\infty} h(u_1', u_2')\, x(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2'$$

see convolution_demo.m....

Linear Systems: convolution theorem

Now take the 2-d Fourier transform of previous equation

$$Y(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

$$= \iint_{-\infty}^{\infty} \left[\iint_{-\infty}^{\infty} h(u_1', u_2')\, x(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2' \right] e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2$$

Interchange order of integration on RHS and use spatial shift theorem

$$= \iint_{-\infty}^{\infty} h(u_1', u_2') \left[\iint_{-\infty}^{\infty} x(u_1 - u_1', u_2 - u_2')\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2 \right] du_1'\, du_2'$$

$$= \iint_{-\infty}^{\infty} h(u_1', u_2') \left[e^{-j(\omega_1 u_1' + \omega_2 u_2')}\, X(\omega_1, \omega_2) \right] du_1'\, du_2'$$

$$\therefore\ Y(\omega_1, \omega_2) = X(\omega_1, \omega_2) \iint_{-\infty}^{\infty} h(u_1', u_2')\, e^{-j(\omega_1 u_1' + \omega_2 u_2')}\, du_1'\, du_2'$$

Linear Systems: convolution theorem cont...

The integral term is the FT of the impulse response, ie the system frequency response $H(\omega_1, \omega_2)$.

$$\therefore \quad \boxed{Y(\omega_1, \omega_2) = X(\omega_1, \omega_2)\, H(\omega_1, \omega_2)}$$

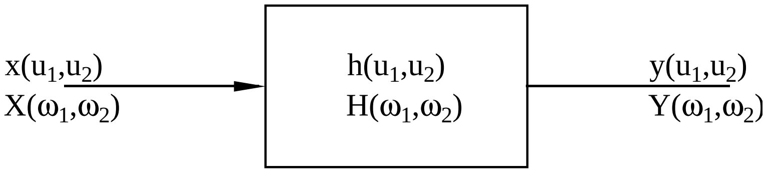
Thus, the FT of the output of any linear system can be obtained from FT of the input and the frequency response. If $*$ denotes the convolution operation and

$$y(u_1, u_2) = f(u_1, u_2) * g(u_1, u_2)$$

then we know that the Fourier transforms of these functions are related multiplicatively, i.e.

$$Y(\omega_1, \omega_2) = F(\omega_1, \omega_2)\, G(\omega_1, \omega_2)$$

Summary of LSI system relationships



The input-output relationships for a linear spatially-invariant (LSI) system can be summarised as:

Spatial Domain

$$y(u_1, u_2) = \iint_{-\infty}^{\infty} x(u_1', u_2')\, h(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2'$$

$$= \iint_{-\infty}^{\infty} h(u_1', u_2')\, x(u_1 - u_1', u_2 - u_2')\, du_1'\, du_2'$$

Frequency Domain

$$Y(\omega_1, \omega_2) = X(\omega_1, \omega_2)\, H(\omega_1, \omega_2)$$

where:

$$X(\omega_1, \omega_2) = \iint_{-\infty}^{\infty} x(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2 \qquad \text{(Input spectrum)}$$

$$x(u_1, u_2) = \frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} X(\omega_1, \omega_2)\, e^{j(\omega_1 u_1 + \omega_2 u_2)}\, d\omega_1\, d\omega_2 \qquad \text{(Input signal)}$$

Similarly for the output image $y(u_1, u_2)$ and output spectrum $Y(\omega_1, \omega_2)$.

$$H(\omega_1, \omega_2) = \iint_{-\infty}^{\infty} h(u_1, u_2)\, e^{-j(\omega_1 u_1 + \omega_2 u_2)}\, du_1\, du_2 \qquad \text{(Frequency response)}$$

$$h(u_1, u_2) = \frac{1}{(2\pi)^2} \iint_{-\infty}^{\infty} H(\omega_1, \omega_2)\, e^{j(\omega_1 u_1 + \omega_2 u_2)}\, d\omega_1\, d\omega_2 \qquad \text{(Impulse response)}$$

J. Lasenby (2016)

IIB 4F8: Image Processing and Image Coding

Handout 2: Digitisation, Sampling & Quantisation and the Discrete 2D Fourier Transform

J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

Image Digitisation

Most image processing is performed with a digital system; we must therefore convert images into a finite set of sample values (sampling) and each sample value must be represented as a finite precision number (quantisation). We model digitized images as *bandlimited* signals – a fairly good approximation to the truth in most cases. A function $f(u_1, u_2)$ is said to be bandlimited if its Fourier transform, $F(\omega_1, \omega_2)$, is zero outside some closed region in the frequency plane: e.g.

$$F(\omega_1, \omega_2) = 0 \text{ if } |\omega_1| > \omega_{1c} \text{ and } |\omega_2| > \omega_{2c}$$

ω_{1c} and ω_{2c} are the u_1 and u_2 bandwidths of the image. If the FT is circularly symmetric, such that $F(\omega_1, \omega_2) = 0$ for $|\boldsymbol{\omega}| > \omega_c$ (where $|\boldsymbol{\omega}|^2 = \omega_1^2 + \omega_2^2$), then ω_c is termed the *bandwidth*.

Sampling

The purpose of sampling is to convert a 2-d continuous signal into an array of sample values from the continuous signal. There are a number of possible different grids which can be used as the basis for sampling: we discuss 2 of these.

Rectangular sampling

Consider the following 2-d sampling function

$$s(u_1, u_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta(u_1 - n_1\Delta_1, u_2 - n_2\Delta_2)$$

which corresponds to a uniform rectangular grid of impulse functions spaced at intervals of Δ_1 in the u_1 direction and intervals of Δ_2 in the u_2 direction, as shown in figure 1.

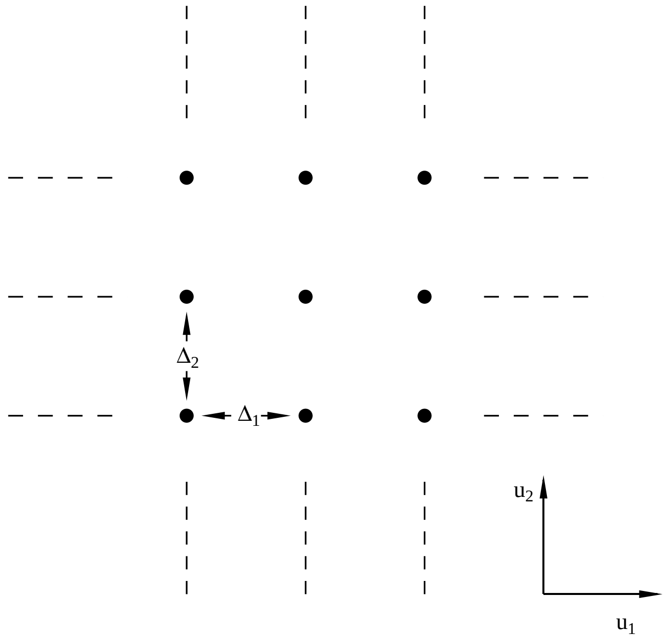


Figure 1: Rectangular sampling grid

Sampling on a rectangular grid

The sampled image $g_s(u_1, u_2)$ is given by:

$$g_s(u_1, u_2) = s(u_1, u_2)\, g(u_1, u_2)$$

where $g(u_1, u_2)$ is the continuous 2d function. s is periodic in u_1 and u_2 directions, with periods Δ_1 and Δ_2. Can therefore write as a 2d Fourier series. Recall that a 1d periodic function $f(x)$ (period T) can be written as

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 x} \quad \text{with} \quad \omega_0 = \frac{2\pi}{T}$$

Similarly we write the 2d Fourier series for $s(u_1, u_2)$ as

$$s(u_1, u_2) = \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} c(p_1, p_2)\, e^{j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)} \qquad (1)$$

where: $\Omega_1 = \frac{2\pi}{\Delta_1}$ and $\Omega_2 = \frac{2\pi}{\Delta_2}$

Fourier Coefficients of s

Recall, in 1D the Fourier coefficients are given by

$$c_n = \frac{1}{T}\int_{\alpha}^{\alpha+T} f(x) e^{-j\omega_0 n x} dx$$

We find that the 2d Fourier coefficients, $c(p_1, p_2)$ are given by

$$c(p_1, p_2) = \frac{1}{\Delta_1\, \Delta_2}\int_{-\frac{\Delta_2}{2}}^{\frac{\Delta_2}{2}} \int_{-\frac{\Delta_1}{2}}^{\frac{\Delta_1}{2}} s(u_1, u_2)\, e^{-j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}\, du_1\, du_2$$

Now substitute for the sampling function $s(u_1, u_2)$:

$$c(p_1, p_2) = \frac{1}{\Delta_1\, \Delta_2}\int_{-\frac{\Delta_2}{2}}^{\frac{\Delta_2}{2}} \int_{-\frac{\Delta_1}{2}}^{\frac{\Delta_1}{2}} \left[\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta(u_1 - n_1\Delta_1, u_2 - n_2\Delta_2)\right]$$

$$\times e^{-j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}\, du_1\, du_2$$

$$\Longrightarrow \quad c(p_1, p_2) = \frac{1}{\Delta_1\, \Delta_2} \quad \text{for all} \quad p_1, p_2$$

FT of sampled signal

The sampled image may then be expressed as:

$$g_s(u_1,u_2) = g(u_1,u_2)\,\frac{1}{\Delta_1\,\Delta_2}\sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} e^{j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}$$

Using the frequency shift or spatial modulation theorem to take the Fourier transform

$$g(u_1,u_2)e^{j(p_1\Omega_1 u_1+p_2\Omega_2 u_2)} \Leftrightarrow G(\omega_1-\Omega_1 p_1, \omega_2-\Omega_2 p_2)$$

gives:

$$\boxed{G_s(\omega_1,\omega_2) = \frac{1}{\Delta_1\,\Delta_2}\sum_{p_1=-\infty}^{\infty}\sum_{p_2=-\infty}^{\infty} G(\omega_1 - p_1\Omega_1, \omega_2 - p_2\Omega_2)}$$

It can therefore be seen that the Fourier transform or spectrum of the sampled 2d signal is the periodic repetition of the spectrum of the unsampled 2d signal (centred on the ‘grid’ points in frequency space) – precisely analogous to the 1d case.

Example

Figures show amplitude of the spectrum of a 2d signal plotted as a mesh plot and as a gray-scale plot – the spectrum here is a truncated Gaussian (therefore implying a bandlimited signal).

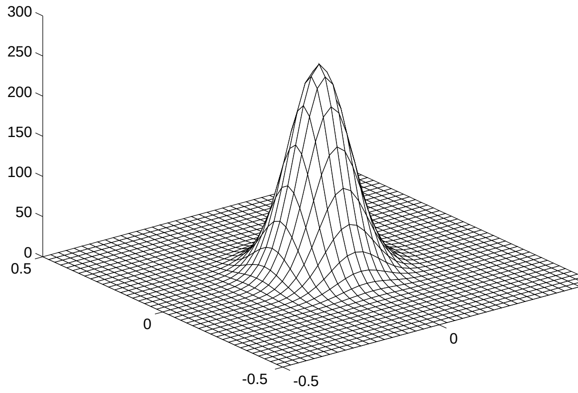


Figure 2: Spectrum of a 2D signal: contour

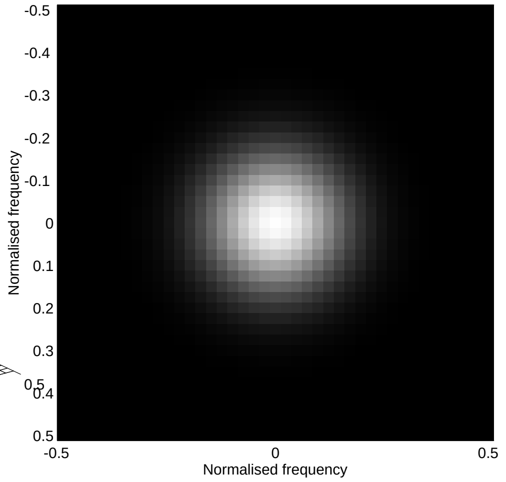


Figure 3: Spectrum of a 2D signal: greyscale

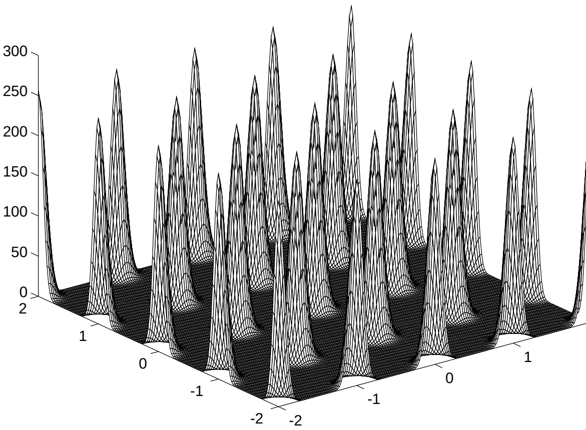


Figure 4: Spectrum $G_s(\omega_1, \omega_2)$ of sampled 2-dimensional signal

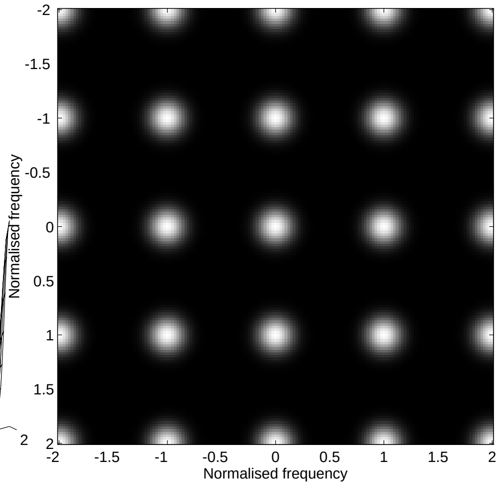


Figure 5: Spectrum $G_s(\omega_1, \omega_2)$ of sampled 2-dimensional signal (greyscale)

Nyquist frequencies

If the image is spatially bandlimited to Ω_{B1} and Ω_{B2} then the original continuous image can be recovered from the sampled image by ideal low-pass filtering at Ω_{B1}, Ω_{B2} if the samples are taken such that $\Omega_{B1} < \frac{1}{2}\Omega_1$ and $\Omega_{B2} < \frac{1}{2}\Omega_2$ so that the periodic repeats of the spectrum do not overlap – this can also be written as:

$$\frac{2\pi}{\Delta_1} > 2\Omega_{B1} \qquad \frac{2\pi}{\Delta_2} > 2\Omega_{B2}$$

$2\Omega_{B1}$ and $2\Omega_{B2}$ are known as the *2d Nyquist frequencies*. Thus the *2d sampling theorem* states that a bandlimited image sampled at or above its u_1 and u_2 Nyquist rates can be recovered without error by low-pass filtering the sampled spectrum.

Sampling below the Nyquist rates causes *aliasing* to occur, where we have overlap of the repeated unsampled spectra in the frequency plane. If aliasing occurs we are not able to recover the true spectrum without error.

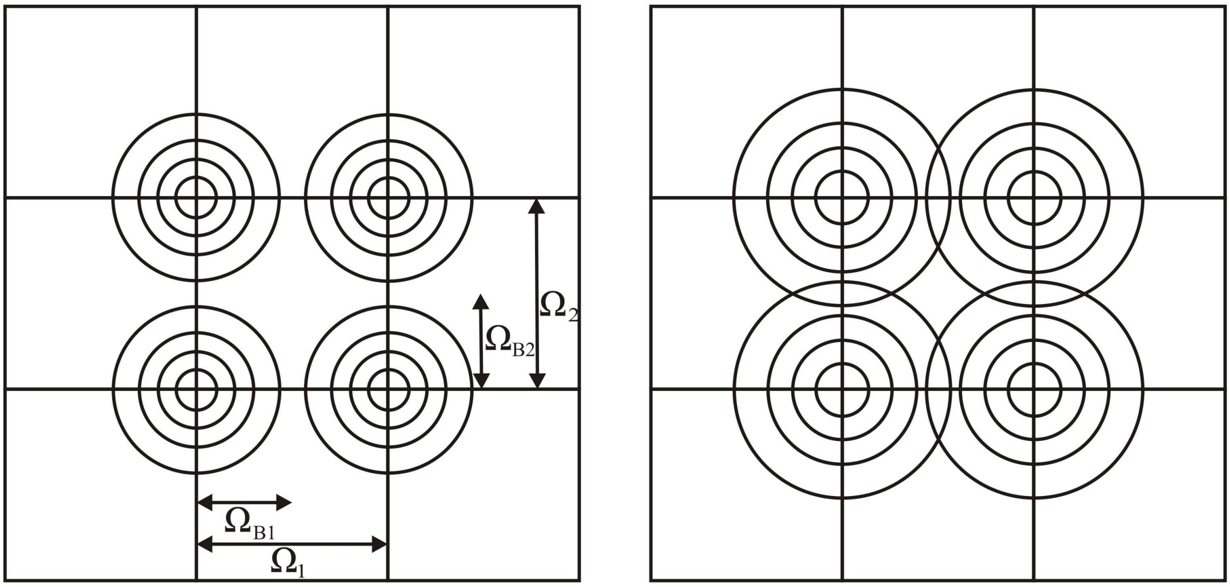
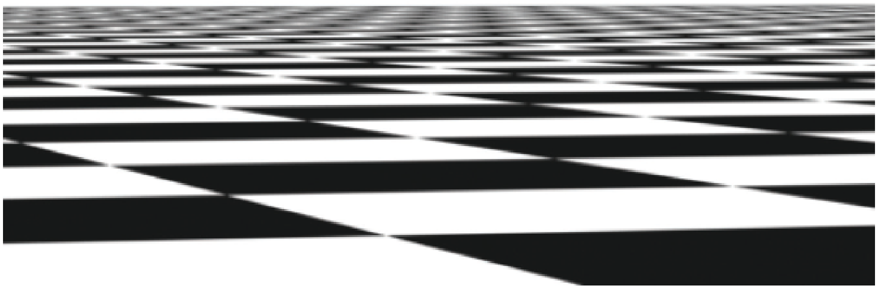
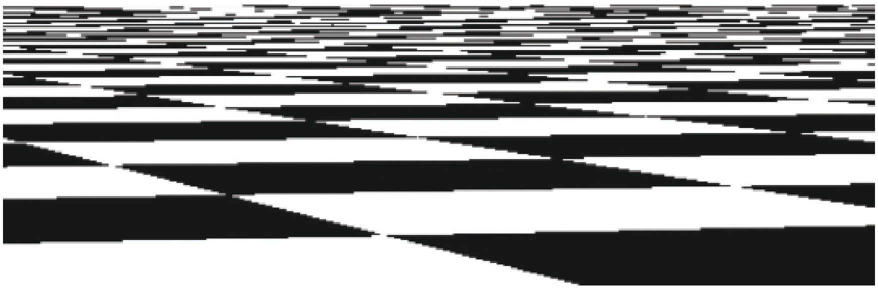


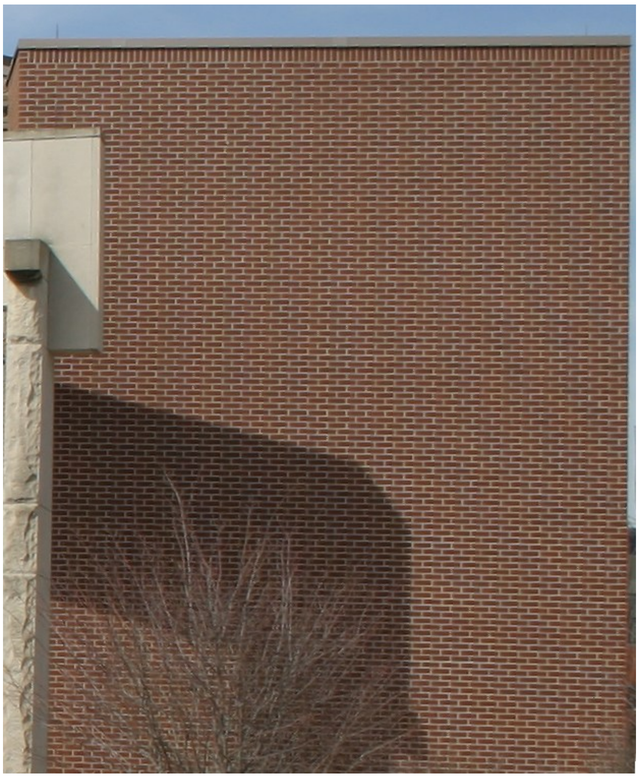
Figure 6: Illustration of 2D aliasing

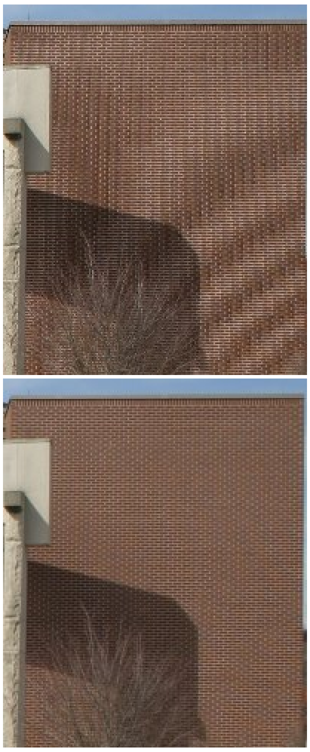
Example of Aliasing - I





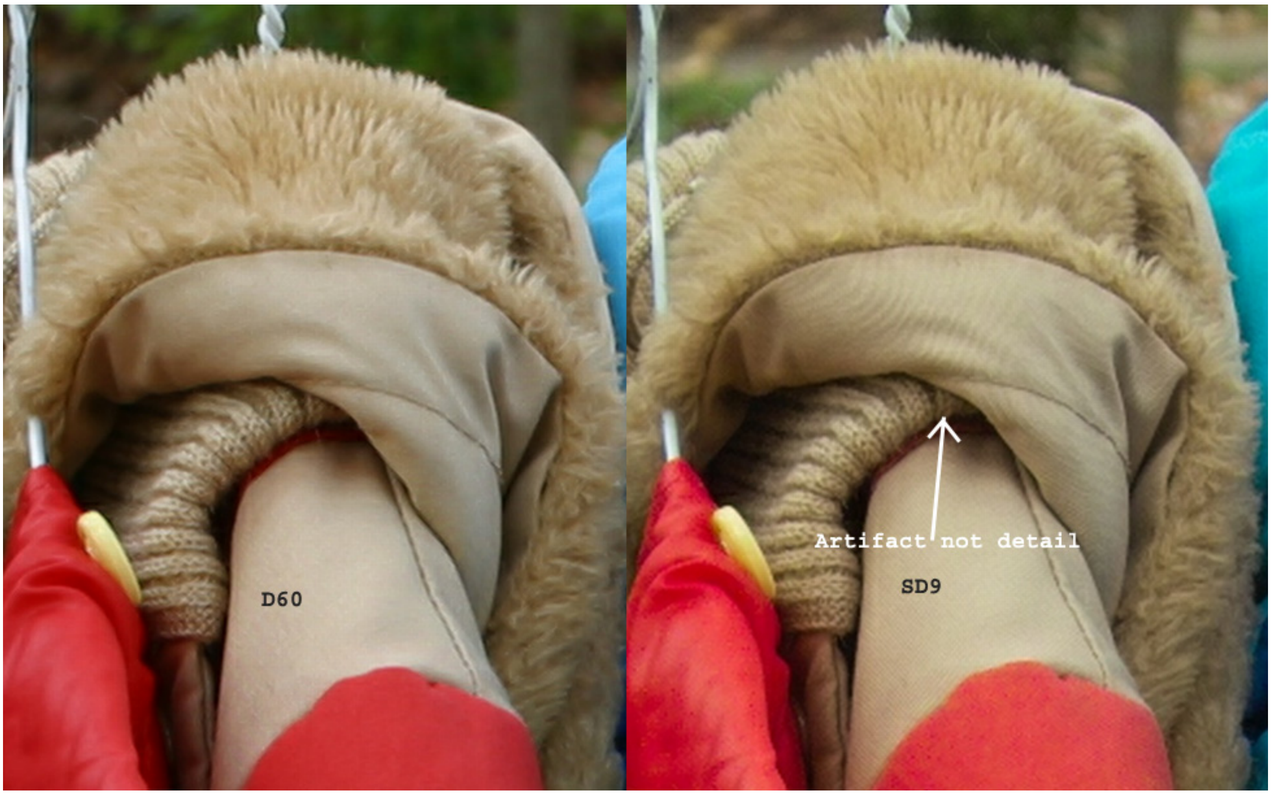
Example of Aliasing - II





Moire

Example of anti-aliasing in photography



Recovering original signal

For sampling at or above the Nyquist frequency, we can recover the unsampled spectrum exactly by low-pass filtering with $H(\omega_1, \omega_2)$ given by

$$H(\omega_1, \omega_2) = \Delta_1 \Delta_2 \quad \text{if } \omega_1, \omega_2 \in R, \quad 0 \quad \text{otherwise}$$

where R is any area in the frequency plane completely containing the spectrum of the unsampled signal – most usually we would take R as $[-\frac{\pi}{\Delta_i} < \omega_i < \frac{\pi}{\Delta_i}]$ (since $\frac{\pi}{\Delta_i} > \Omega_{Bi}$).

The FT of our unsampled signal is thus

$$G(\omega_1, \omega_2) = H(\omega_1, \omega_2) G_s(\omega_1, \omega_2)$$

Multiplication in the spatial domain implies convolution in the frequency domain, and vice versa; we therefore have (where h is the inverse FT of H)

$$g(u_1, u_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u_1 - u_1', u_2 - u_2') g_s(u_1', u_2') du_1' du_2'$$

Since we can write g_s as

$$g_s(u_1', u_2') = \sum_n \sum_m g(n\Delta_1, m\Delta_2) \delta(u_1' - n\Delta_1, u_2' - m\Delta_2)$$

we can use this to write our signal g as

$$
\begin{aligned}
g(u_1, u_2) &= \sum_n \sum_m \iint_{-\infty}^{\infty} h(u_1 - u_1', u_2 - u_2') g(n\Delta_1, m\Delta_2) \delta(u_1' - n\Delta_1, u_2' - m\Delta_2) du_1' du_2' \\
&= \sum_n \sum_m h(u_1 - n\Delta_1, u_2 - m\Delta_2) g(n\Delta_1, m\Delta_2)
\end{aligned}
$$

We can show that h, the inverse FT of H is [Exercise]:

$$h(u_1, u_2) = \text{sinc}\, \frac{u_1 \pi}{\Delta_1} \text{sinc}\, \frac{u_2 \pi}{\Delta_2}$$

Thus we can now write our original image in terms of the samples and the above sinc functions

$$\boxed{g(u_1, u_2) = \sum_n \sum_m g(n\Delta_1, m\Delta_2)\ \text{sinc}\, \frac{(u_1 - n\Delta_1)\pi}{\Delta_1}\ \text{sinc}\, \frac{(u_2 - m\Delta_2)\pi}{\Delta_2}}$$

We are effectively using these sinc functions to *interpolate* between the samples.
You should recognise this result from 1D sampling theory, where the *reconstruction theorem*, for signals sampled at or above the Nyquist frequency, was cast in the form of a sum of interpolations using sinc functions. It is intuitively obvious that 2D signals should behave in the same way.
So, what we have derived here is fundamental to 2D sampling: **provided a signal is sampled at or above its 2D Nyquist frequencies, we are able to perfectly reconstruct the signal from its samples**.
Of course, we could have chosen other sampling patterns. Different patterns may have advantages under some circumstances and we discuss another such pattern in the following section.

Diamond Sampling Grid

The Diamond sampling scheme which is used in a number of applications is shown below:

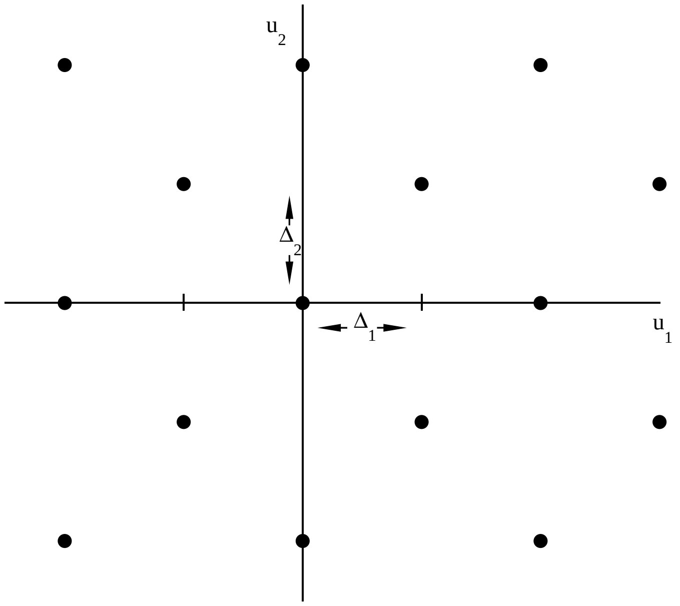


Figure 7: Diamond sampling grid

Diamond Sampling Grid

Represent the sampling grid by two separate 2d sampling functions as illustrated below

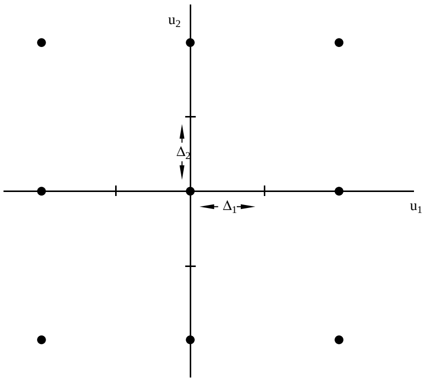


Figure 8: $s_1(u_1, u_2)$

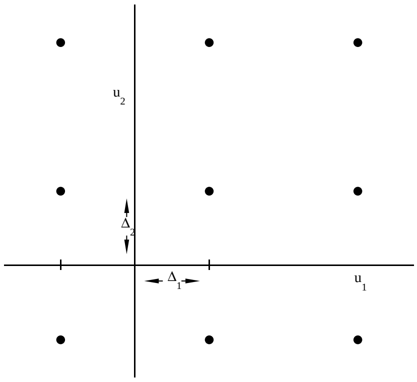


Figure 9: $s_2(u_1, u_2)$

$$s_1(u_1, u_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta[u_1 - 2\,n_1\Delta_1, u_2 - 2\,n_2\Delta_2]$$

$$s_2(u_1, u_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \delta[u_1 - (2\,n_1 + 1)\,\Delta_1, u_2 - (2\,n_2 + 1)\,\Delta_2]$$

Since s_1 and s_2 are periodic (with periods $2\Delta_1$ and $2\Delta_2$) we can represent them as 2d Fourier series.

$$s_1(u_1, u_2) = \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} c_1(p_1, p_2)\, e^{j(p_1 \frac{\Omega_1}{2} u_1 + p_2 \frac{\Omega_2}{2} u_2)}$$

$$s_2(u_1, u_2) = \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} c_2(p_1, p_2)\, e^{j(p_1 \frac{\Omega_1}{2} u_1 + p_2 \frac{\Omega_2}{2} u_2)}$$

$$\text{where} \quad \Omega_1 = \frac{2\pi}{\Delta_1} \quad \text{and} \quad \Omega_2 = \frac{2\pi}{\Delta_2}$$

Thus the sampled function can be written as

$$g_s(u_1, u_2) = g(u_1, u_2)[s_1(u_1, u_2) + s_2(u_1, u_2)]$$

$$= \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} [c_1(p_1, p_2) + c_2(p_1, p_2)]\, g(u_1, u_2)\, e^{j(p_1 \frac{\Omega_1}{2} u_1 + p_2 \frac{\Omega_2}{2} u_2)}$$

Take the 2-D Fourier transform using the frequency shift theorem to give:

$$G_s(\omega_1, \omega_2) = \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} [c_1(p_1, p_2) + c_2(p_1, p_2)]\, G(\omega_1 - p_1 \frac{\Omega_1}{2}, \omega_2 - p_2 \frac{\Omega_2}{2})$$

By the usual process we can determine the Fourier coefficients

$$c_1(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \int_{-\Delta_2}^{\Delta_2} \int_{-\Delta_1}^{\Delta_1} s_1(u_1, u_2)\, e^{-j(p_1 \frac{\Omega_1}{2} u_1 + p_2 \frac{\Omega_2}{2} u_2)}\, du_1\, du_2 \tag{2}$$

$$c_2(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \int_{-\Delta_2}^{\Delta_2} \int_{-\Delta_1}^{\Delta_1} s_2(u_1, u_2)\, e^{-j(p_1 \frac{\Omega_1}{2} u_1 + p_2 \frac{\Omega_2}{2} u_2)}\, du_1\, du_2 \tag{3}$$

Clearly the factor $\frac{1}{4\,\Delta_1\,\Delta_2}$ comes from $\frac{1}{T_1 T_2}$ with $T_i = 2\Delta_i$.

Substitute for $s_1(u_1, u_2)$ in equation 2 and interchange integral and summation operations to give:

$$c_1(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \int_{-\Delta_2}^{\Delta_2} \int_{-\Delta_1}^{\Delta_1} \delta[u_1 - 2\,n_1\Delta_1, u_2 - 2\,n_2\Delta_2]$$

$$\times\; e^{-j(p_1\frac{\Omega_1}{2}u_1 + p_2\frac{\Omega_2}{2}u_2)}\,du_1\,du_2$$

Since the only contribution to the integral will come when $n_1 = n_2 = 0$, we have

$$\therefore\;\; c_1(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2}$$

Similarly, substituting for $s_2(u_1, u_2)$ in equation 3 and interchanging integral and summation operations gives

$$c_2(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \int_{0}^{2\Delta_2} \int_{0}^{2\Delta_1} \delta[u_1 - (2\,n_1 + 1)\,\Delta_1, u_2 - (2\,n_2 + 1)\,\Delta_2]$$

$$\times\; e^{-j(p_1\frac{\Omega_1}{2}u_1 + p_2\frac{\Omega_2}{2}u_2)}\,du_1\,du_2$$

note here that we change our limits to take the period between $[0, 2\Delta_1]$ and $[0, 2\Delta_2]$ (recall that we can take any interval covering a whole period), so that the only contributions to the integral now come from $n_1 = 0$ and $n_2 = 0$ giving $u_1 = \Delta_1$ and $u_2 = \Delta_2$. Since $(p_i\Omega_i\Delta_i)/2 = p_i\pi$ we have

$$c_2(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} e^{-j(p_1+p_2)\,\pi}$$

Thus

$$c_1(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \tag{4}$$

$$c_2(p_1, p_2) = \frac{1}{4\,\Delta_1\,\Delta_2} e^{-j(p_1+p_2)\,\pi} \tag{5}$$

Substituting for $c_1(p_1, p_2)$ and $c_2(p_1, p_2)$ in the equation for the sampled signal spectrum gives:

$$G_s(\omega_1, \omega_2) = \frac{1}{4\,\Delta_1\,\Delta_2} \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} [1 + e^{-j(p_1+p_2)\,\pi}]\, G(\omega_1 - p_1\frac{\Omega_1}{2}, \omega_2 - p_2\frac{\Omega_2}{2})$$

Since $1 + e^{-j(p_1+p_2)\,\pi} = \begin{cases} 0, & p_1 + p_2 = \text{odd} \\ 2, & p_1 + p_2 = \text{even} \end{cases}$

$$\boxed{G_s(\omega_1, \omega_2) = \frac{1}{2\,\Delta_1\,\Delta_2} \sum_{p_1=-\infty}^{\infty} \sum_{p_2=-\infty}^{\infty} G(\omega_1 - p_1\frac{\Omega_1}{2}, \omega_2 - p_2\frac{\Omega_2}{2}) \quad \text{for } p_1 + p_2 \text{ even}} \tag{6}$$

It can be seen that the spectrum of the sampled signal is the periodic repetition of the unsampled signal spectrum, where the unsampled spectrum repeats itself at (even,even) and (odd,odd) intervals of $\Omega_1/2$ and $\Omega_2/2$ as shown in figure 10.

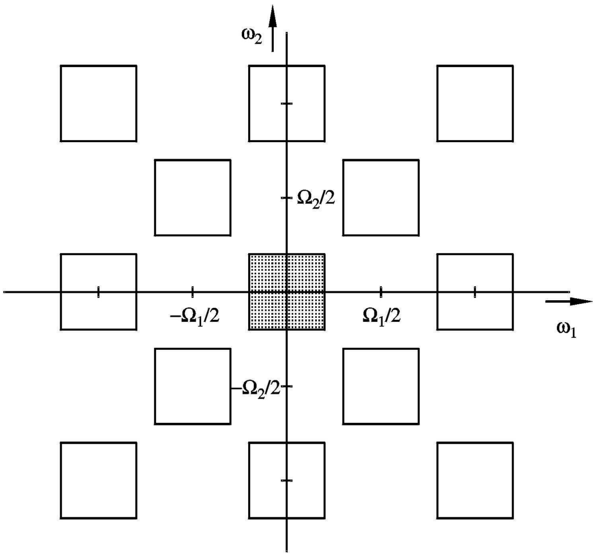


Figure 10: Spectrum of sampled signal with diamond sampling

Sometimes the diamond sampling grid offers an advantage over rectangular sampling as may be seen in figure 11.

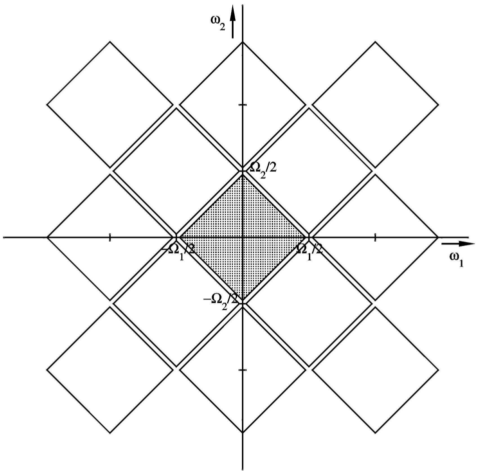


Figure 11: Signal which lends itself to diamond sampling regime

Here the unsampled spectrum can be tightly enclosed by a diamond-shaped region; by sampling on a diamond grid we can then reduce the sampling density by a factor of 2.

The general rule is that if an image does not contain high frequencies simultaneously in both dimensions, then we will be better off sampling on a diamond grid.

For functions which are circularly symmetric or bandlimited over a circular region, it can be shown that sampling on a hexagonal grid is significantly better (requiring fewer samples) than rectangular sampling.

Image Quantisation

Having determined the (spatial) rate at which a 2-D signal is sampled, now look at how many bits are required to represent the amplitude of each sample in a subjectively acceptable manner.

It appears that the eye is significantly less sensitive to quantisation error in images than the sensitivity which the ear exhibits with quantised audio signals.

The generally accepted criterion for image quantisation is a resolution of 8 bits (0 to $[2^8 - 1] = 255$ or 1 to 256) whereas 16 bits are required for high quality audio.

Although 8 bits gives very adequate representation for images, 10 (0 to $[2^{10} - 1] = 1023$ or 1 to 1024) or 12 (0 to $[2^{12} - 1] = 4095$ or 1 to 4096) bits are sometimes used for the very highest quality material.

Figures 12 to 15 show an image with successively coarser quantisation.

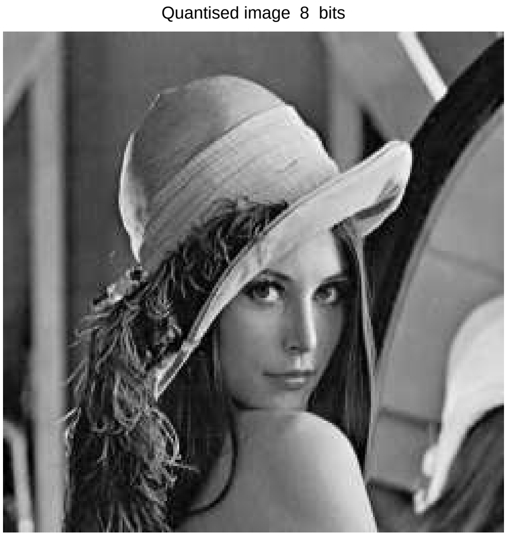


Figure 12: Image quantised to 8 bits: 256 grey levels (0 to 255)

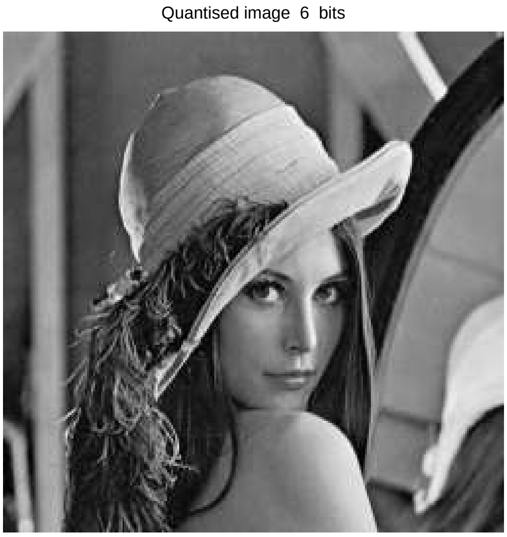


Figure 13: Image quantised to 6 bits: 64 grey levels (0 to 63)

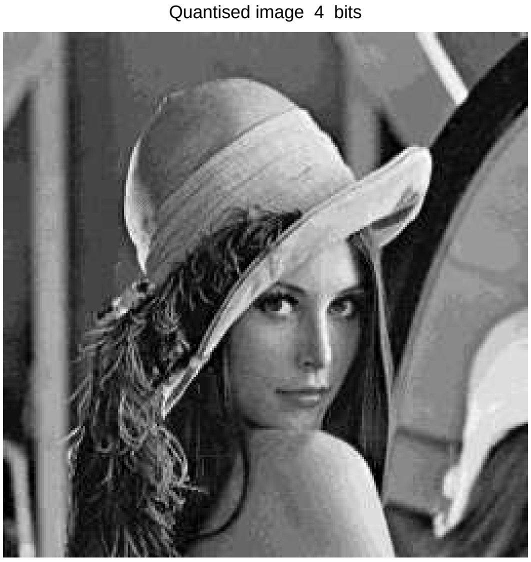


Figure 14: Image quantised to 4 bits: 16 grey levels (0 to 15)

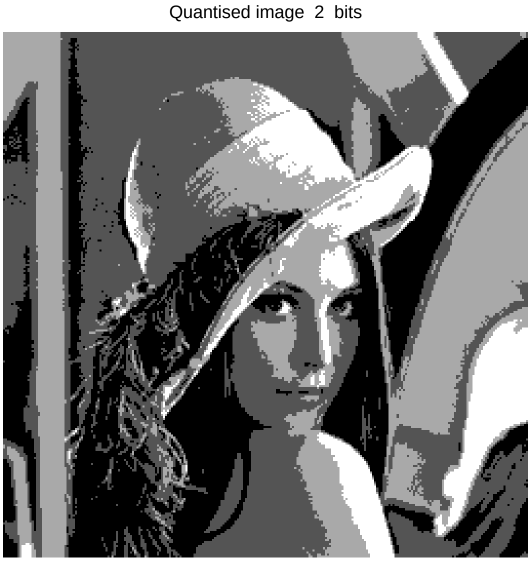


Figure 15: Image quantised to 2 bits: 4 grey levels (0 to 3)

The 2d Discrete Fourier Transform (DFT)

Have seen how digital images can be represented as sampled versions of a continuous 2d function and how the Fourier transform (FT) of this sampled image is the periodic repetition of the spectrum of the original signal.

We now ask how to deal with real-life cases when we have a finite 2d array of numbers (constituting our image) and we want to obtain the best approximation to the FT of this image given by a finite array of the same size (so the two are fully invertible). Recall that in 1d we periodically extended our finite set of N samples, $\{f(nT_s), n = 0, .., N-1\}$ as shown in figure 16;

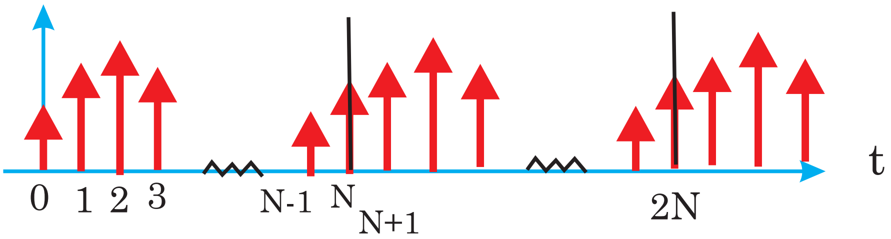


Figure 16: Periodic repetition of the finite set of samples, $f_s(t)$

Revision 1d DFT

We know that the FT of a periodic function takes discrete values and that the FT of a sampled signal repeats every interval of the sampling frequency: $\Longrightarrow$ the FT of the above periodic extension will be periodic (period NT_s) and defined at discrete frequency values. We therefore get precisely what we want and are able to define the 1d DFT and the 1d inverse DFT as:

$$F_k = \sum_{n=0}^{N-1} f_n e^{-j2\pi\frac{nk}{N}} \qquad 0 \leq k \leq N-1 \tag{7}$$

$$f_n = \frac{1}{N}\sum_{k=0}^{N-1} F_k e^{j2\pi\frac{nk}{N}} \qquad 0 \leq n \leq N-1 \tag{8}$$

where we start with a set of N samples, $f_n = f(nT_s)$ and produce a set of N frequency samples, $F_k = F(k\omega_0)$ where $\omega_0 = 2\pi/T$ and $T = NT_s$.

The 2d DFT

To get the 2d DFT, again we periodically extend our image to tile the plane as shown in figure 17:

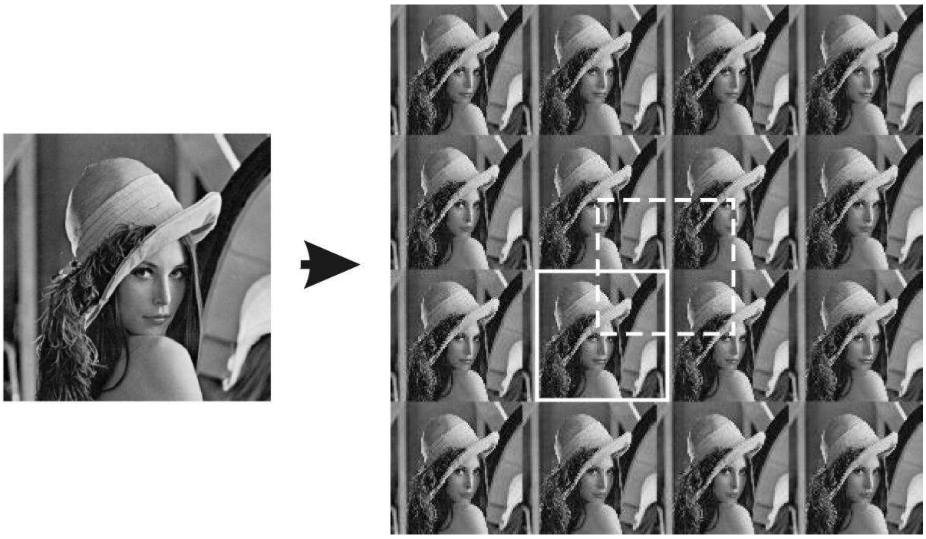


Figure 17: Periodic repetition of the Lenna image. The solid and dotted white squares show two possible placings of the origin of the tiling

In 1d could take our period from any α to $\alpha + T$; in 2d can choose any relevant part of the plane – Figure 17 shows the two most common cases.

Solid rectangle shows the original image, dotted rectangle shows an offset section where the 1st & 3rd and the 2nd & 4th quadrants have been interchanged. This interchange is performed by the command `fftshift` in Matlab.

For the image $f(u_1, u_2)$, we can write the $N_1 \times N_2$ sampled image as

$$f_s(u_1, u_2) = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} f(u_1, u_2)\delta(u_1 - n_1\Delta_1, u_2 - n_2\Delta_2)$$

Let $\bar{f}_s$ be the periodic extension of f_s tiling the whole 2d plane. $\bar{f}_s$ periodic $\Longrightarrow$ write it as a Fourier series

$$\bar{f}_s(u_1, u_2) = \sum_{p_2=-\infty}^{\infty} \sum_{p_1=-\infty}^{\infty} c(p_1, p_2)e^{j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}$$

where $\Omega_i = 2\pi/T_i$, $T_i = N_i\Delta_i$. The Fourier coefficients are given by

$$\begin{aligned} c(p_1, p_2) &= \frac{1}{T_1 T_2}\int_{-T_2/2}^{+T_2/2}\int_{-T_1/2}^{+T_1/2} f_s(u_1, u_2)e^{-j(p_1\Omega_1 u_1 + p_2\Omega_2 u_2)}du_1 du_2 \\ &= \frac{1}{T_1 T_2}\sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} f(n_1\Delta_1, n_2\Delta_2)e^{-j(p_1\Omega_1 n_1\Delta_1 + p_2\Omega_2 n_2\Delta_2)} \end{aligned}$$

Note that we can associate $c(p_1, p_2)T_1 T_2$ with the Fourier transform at $(p_1\Omega_1, p_2\Omega_2)$ [recall the 1D case where kth fourier coefficient of a sampled, periodic function f_s was equivalent to $\frac{1}{T}F(k\omega_0)$, where F_s is the FT of f_s and $\omega_0 = \frac{2\pi}{T}$].

Now, using $\Delta_i\Omega_i = 2\pi/N_i$, we are able to arrive at our definition of the 2d DFT: $F_{p_1 p_2}$ denotes the FT at frequencies $(p_1\Omega_1, p_2\Omega_2)$ and $f_{n_1 n_2}$ the image values at $(n_1\Delta_1, n_2\Delta_2)$;

$$\boxed{F_{p_1,p_2} = \sum_{n_2=0}^{N_2-1}\sum_{n_1=0}^{N_1-1} f_{n_1 n_2} e^{-2\pi j(p_1\frac{n_1}{N_1} + p_2\frac{n_2}{N_2})}} \qquad (9)$$

the inverse DFT (IDFT) is similarly given by

$$\boxed{f_{n_1,n_2} = \frac{1}{N_1 N_2} \sum_{p_2=0}^{N_2-1} \sum_{p_1=0}^{N_1-1} F_{p_1 p_2} e^{2\pi j (n_1 \frac{p_1}{N_1} + n_2 \frac{p_2}{N_2})}} \tag{10}$$

$\therefore$ we can transform an image using the 2d DFT – note that the form of this transform means that we can do it as 2 1d DFTs – giving an array of complex numbers which gives the Fourier transform of that image.

2d DFT cont...

If we shift the image before transforming as shown below,

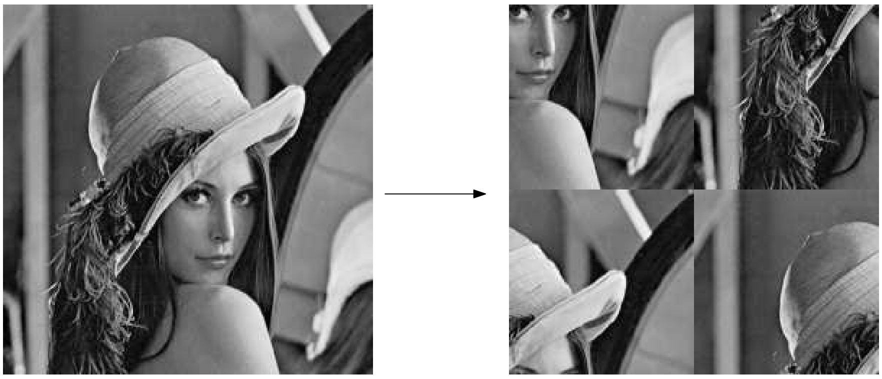


Figure 18: Original image and shifted image with 1st & 3rd and 2nd & 4th quadrants interchanged

then the effect of this is to put the ‘dc level’ (pixel $(1, 1)$) at the centre of the image. In most cases, it does not matter if we shift or do not shift, but it is important to understand what effect changing the position of the dc level has on the DFT.

Effects of shifting

A shift in the spatial domain implies multiplication by a complex exponential in the frequency domain. Thus if we shift our image (modulo N_1 and N_2) the 2D-DFT is given by

$$f_{n_1+\frac{N_1}{2},n_2+\frac{N_2}{2}} = \frac{1}{N_1 N_2}\sum_{p_2=0}^{N_2-1}\sum_{p_1=0}^{N_1-1} F_{p_1 p_2} e^{2\pi j([n_1+\frac{N_1}{2}]\frac{p_1}{N_1}+[n_2+\frac{N_2}{2}]\frac{p_2}{N_2})}$$

which we can rearrange to give

$$f_{n_1+\frac{N_1}{2},n_2+\frac{N_2}{2}} = \frac{1}{N_1 N_2}\sum_{p_2=0}^{N_2-1}\sum_{p_1=0}^{N_1-1} \left(F_{p_1 p_2} e^{\pi j p_1+\pi j p_2}\right) e^{2\pi j(n_1\frac{p_1}{N_1}+n_2\frac{p_2}{N_2})}$$

Thus, in the frequency domain we are applying a ± 1 factor across the image (since $e^{\pi j p_i}$ changes from $+1$ to -1 moving from even to odd pixels). A one pixel shift, ie f_{n_1,n_2+1}, gives an FT of $F_{p_1 p_2} e^{2\pi j\frac{p_2}{N_2}}$, giving a phase ramp across the image.

Effects of shifting – visualisation

Consider visualising our frequency data. Take a centred 2d gaussian – Both the original image and the shifted image (`fftshift`) have a Fourier transform which gives the DC component ($(0,0)$ frequencies) in the top left-hand corner of the image. If we want to visualise this so that the $(0,0)$ frequency component is in the centre of the image then we `fftshift` our transform.

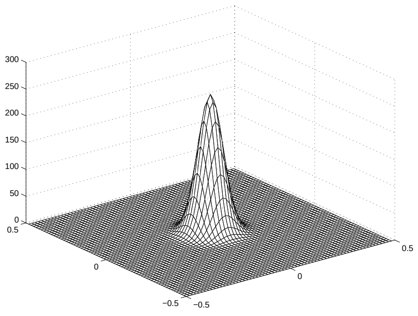


Figure 19: Gaussian centred in the image

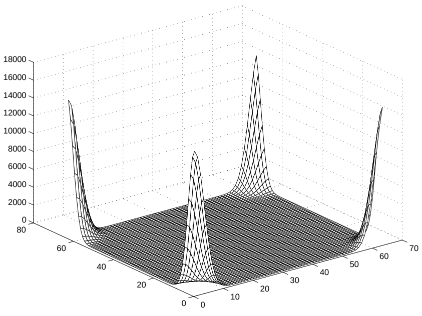


Figure 20: 2D-DFT of gaussian

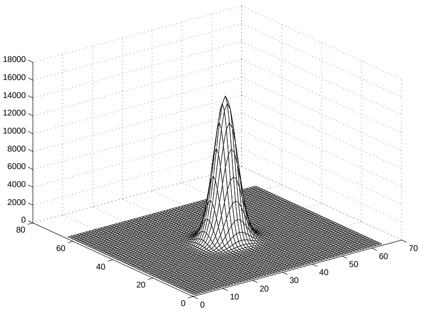
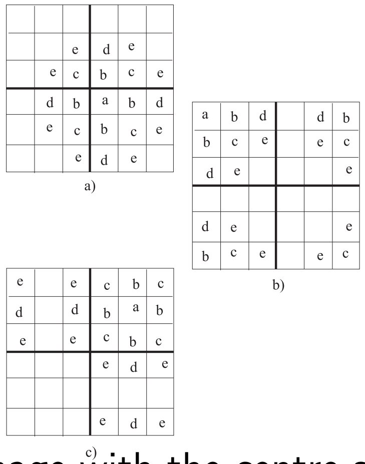


Figure 21: `fftshift` applied to 2D-DFT

Exercise

Here we see the same symmetric arrangement shifted to have its centre at a different part of the image. Investigate the 2d DFT of such an image, in particular, note how the phase of the DFT differs between the three examples.



		e	d	e	
	e	c	b	c	e
	d	b	a	b	d
	e	c	b	c	e
		e	d	e	

a)

a	b	d		d	b
b	c	e		e	c
d	e				e
d	e				e
b	c	e		e	c

b)

e		e	c	b	c
d		d	b	a	b
e		e	c	b	c
			e	d	e
			e	d	e

c)

Figure 22: A symmetric image with the centre shifted to different parts of the image plane. a) centre at position $(N/2+1, N/2+1)$ of the grid, b) centre at position $(1,1)$ of the grid and c) centre at position $(N/2+2, N/2-1)$ of the grid

J. Lasenby (2016)

IIB 4F8: Image Processing and Image Coding

Handout 3: 2D Digital Filters

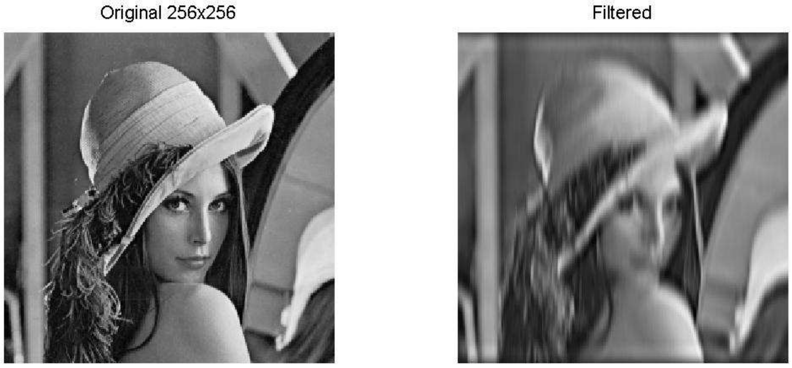
J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

Image Filters

In many applications of multi-dimensional signal processing it is necessary to apply a spatial filtering operation. In some cases 3-D filtering is required, operating in the 2 spatial coordinates and in time. However, at this stage, consideration will be given only to 2D spatial filtering.


Original 256x256

Filtered

Figure 1: Original and filtered (convolved) versions of Lenna

2d Digital Filters

Have seen that we can write the output, $y(u_1, u_2)$, of a linear system (with impulse response $h(u_1, u_2)$) and input $x(u_1, u_2)$, as

$$y(u_1, u_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(u'_1, u'_2)\, x(u_1 - u'_1, u_2 - u'_2) du'_1 du'_2$$

with $Y(\omega_1, \omega_2) = H(\omega_1, \omega_2) X(\omega_1, \omega_2)$. Call this linear system a *2d digital filter* and write the above in terms of discrete sums (supposing a digitised image of infinite extent):

$$y(n_1, n_2) = \sum_{p_2=-\infty}^{\infty} \sum_{p_1=-\infty}^{\infty} h(p_1, p_2)\, x\,(n_1 - p_1, n_2 - p_2)$$

Need to model h as discrete (again, assume infinite extent) $\Longrightarrow$ FT is periodic. Explicity, write the discrete form of h as h_s where

$$h_s(u_1, u_2) = \sum_{n_2=-\infty}^{\infty} \sum_{n_1=-\infty}^{\infty} h(u_1, u_2) \delta(u_1 - n_1 \Delta_1, u_2 - n_2 \Delta_2)$$

Δ_i is the sample spacing in the u_i direction. We can then write its FT, $H_s(\omega_1, \omega_2)$, as

$$\iint_{-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} \sum_{n_1=-\infty}^{\infty} h(u_1, u_2) \delta(u_1 - n_1 \Delta_1, u_2 - n_2 \Delta_2) e^{-j(\omega_1 u_1 + \omega_2 u_2)} du_1 du_2$$

$$= \sum_{n_2=-\infty}^{\infty} \sum_{n_1=-\infty}^{\infty} h(n_1 \Delta_1, n_2 \Delta_2) e^{-j(\omega_1 n_1 \Delta_1 + \omega_2 n_2 \Delta_2)}$$

Note that this is just the discrete time Fourier transform (DTFT).

$H_s(\omega_1, \omega_2)$ is periodic with periods $\Omega_i = 2\pi/\Delta_i$; note then that previous expression is the Fourier Series expansion with $h(n_1\Delta_1, n_2\Delta_2)$ as the Fourier coefficients. Therefore

$$\begin{aligned} h(n_1\Delta_1, n_2\Delta_2) &= \frac{1}{\Omega_1\Omega_2}\int_{-\Omega_2/2}^{\Omega_2/2}\int_{-\Omega_1/2}^{\Omega_1/2} H_s(\omega_1, \omega_2)e^{j(\omega_1 n_1\Delta_1 + \omega_2 n_2\Delta_2)}d\omega_1 d\omega_2 \\ &= \frac{\Delta_1\Delta_2}{(2\pi)^2}\int_{-\pi/\Delta_2}^{\pi/\Delta_2}\int_{-\pi/\Delta_1}^{\pi/\Delta_1} H_s(\omega_1, \omega_2)e^{j(\omega_1 n_1\Delta_1 + \omega_2 n_2\Delta_2)}d\omega_1 d\omega_2 \end{aligned}$$

Letting $\omega_i \rightarrow (\omega_i\Delta_i)$, i.e. use *normalised frequencies*, such that $\Omega_i \rightarrow \frac{2\pi}{\Delta_i}\Delta_i = 2\pi$, we have (putting $\omega_i' = \omega_i\Delta_i$ and sampling intervals of unity):

$$\boxed{\begin{aligned} H_s(\omega_1, \omega_2) &= \sum_{n_2=-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty} h(n_1, n_2)\, e^{-j(\omega_1 n_1 + \omega_2 n_2)} \\ h(n_1, n_2) &= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} H_s(\omega_1, \omega_2)e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2 \end{aligned}}$$

2d Digital Filter relations: summary

$$\boxed{\begin{aligned} H(\omega_1, \omega_2) &= \sum_{n_2=-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty} h(n_1, n_2)\, e^{-j(\omega_1 n_1 + \omega_2 n_2)} \\ h(n_1, n_2) &= \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} H(\omega_1, \omega_2)e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2 \end{aligned}}$$

(where we have dropped the subscript s on H_s). $h(n_1, n_2) = 0$ outside the region of support R_h of the impulse response so that:

$$\boxed{H(\omega_1, \omega_2) = \sum\sum_{(n_1, n_2 \in R_h)} h(n_1, n_2)\, e^{-j(\omega_1 n_1 + \omega_2 n_2)}}$$

Importance of phase in images

In 1-D processing, particularly of audio signals, the phase response (essentially how the phase is altered) of any filtering operation is of relatively minor importance since the auditory system is insensitive to phase.

However, perception of images is very much concerned with lines and edges.

If the filter phase response is non-linear, then the various frequency components which contribute to an edge in an image will be phase shifted with respect to each other in such a way that they no longer add up to produce a sharp edge (i.e. dispersion takes place).

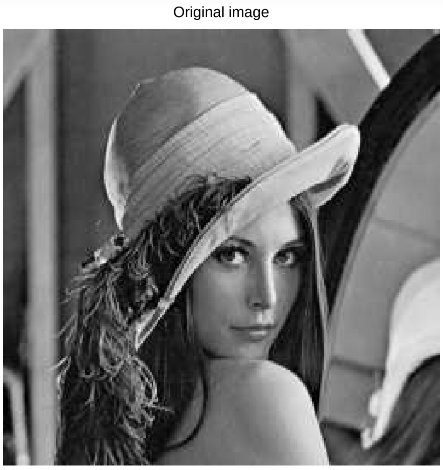


Figure 2: ‘Lenna’ (256× 256) pixels: original image

Take FT of above and
(a) set phases to zero and take IDFT
(b) set amplitudes to 1 and take IDFT

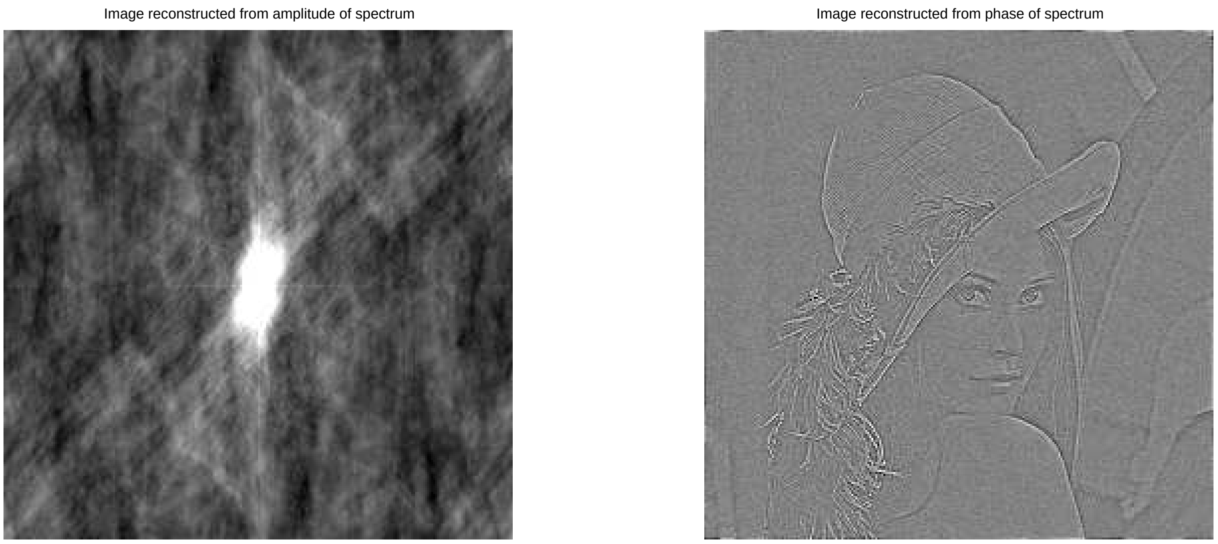


Image reconstructed from amplitude of spectrum

Image reconstructed from phase of spectrum

Figure 3: Amplitude and Phase reconstruction

Zero-phase filters

If a filter frequency response is to be zero-phase then it is purely real so that:

$$H(\omega_1,\omega_2) = H^*(\omega_1,\omega_2)$$

Using our previous expression for H

$$\sum_{(n_1,n_2\in R_h)}\sum h(n_1,n_2)\,e^{-j(\omega_1 n_1+\omega_2 n_2)} = \sum_{(n_1,n_2\in R_h)}\sum h^*(n_1,n_2)\,e^{+j(\omega_1 n_1+\omega_2 n_2)}$$

The above equation is satisfied for all ω_1 and ω_2 if:

$$h^*(n_1,n_2) = h(-n_1,-n_2)$$

Usually only filters with real impulse response coefficients are considered so:

$$h(-n_1, -n_2) = h(n_1, n_2)$$

$\Longrightarrow$ approximately half of the filter parameters are independent variables.

Figure shows the independent parameters for a zero phase filter with a square support region of 5 x 5.

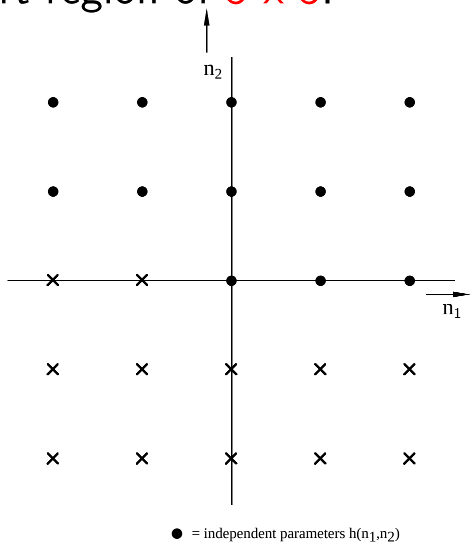


Figure 4: Region of support for 5x5 zero-phase FIR filter

Ideal 2d Filters

Impulse response of ideal 2-D filters can be calculated by taking the 2-D IFT of the *ideal* frequency response. But what might constitute an *ideal* frequency response?

Ideal Rectangular Bandpass Filters

(Assume working with normalised frequencies). Here, the passband of allowed frequencies lies in a rectangular band.

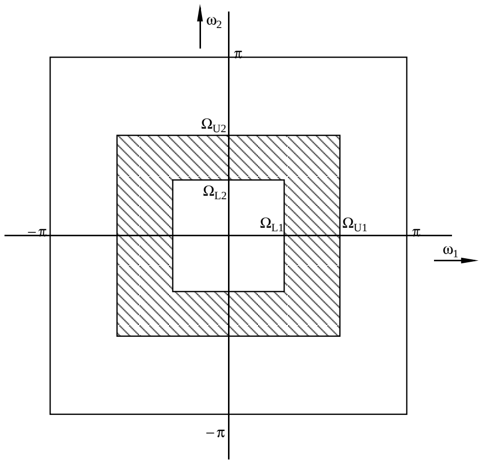


Figure 5: Frequency response of ideal rectangular bandpass filter

See that the frequency response can be expressed as the difference of two lowpass filters, $H = H_u - H_l$ where:

$$H_u(\omega_1, \omega_2) = \begin{cases} 1 & \text{if } |\omega_1| < \Omega_{U1} \text{ and } |\omega_2| < \Omega_{U2} \\ 0 & \text{otherwise} \end{cases}$$

$$H_l(\omega_1, \omega_2) = \begin{cases} 1 & \text{if } |\omega_1| < \Omega_{L1} \text{ and } |\omega_2| < \Omega_{L2} \\ 0 & \text{otherwise} \end{cases}$$

The filter impulse response is given by:

$$h(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2$$

Evaluate this integral by writing $H = H_u - H_l$;

$$h(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\Omega_{U2}}^{\Omega_{U2}} \int_{-\Omega_{U1}}^{\Omega_{U1}} e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2 -$$

$$\frac{1}{(2\pi)^2} \int_{-\Omega_{L2}}^{\Omega_{L2}} \int_{-\Omega_{L1}}^{\Omega_{L1}} e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2$$

$$= \frac{1}{(2\pi)^2} \left[\frac{2\,\sin(\Omega_{U2} n_2)}{n_2} \frac{2\,\sin(\Omega_{U1} n_1)}{n_1} - \frac{2\,\sin(\Omega_{L2} n_2)}{n_2} \frac{2\,\sin(\Omega_{L1} n_1)}{n_1} \right]$$

$$\therefore\; h(n_1, n_2) =$$

$$\boxed{\frac{1}{\pi^2} \left[\Omega_{U2}\, \Omega_{U1}\, \mathrm{sinc}(\Omega_{U2} n_2)\, \mathrm{sinc}(\Omega_{U1} n_1) - \Omega_{L2}\, \Omega_{L1}\, \mathrm{sinc}(\Omega_{L2} n_2)\, \mathrm{sinc}(\Omega_{L1} n_1) \right]}$$

ImageFilter3Demo.m

Note: if we do not work with normalised frequencies but instead take into account the sampling intervals, Δ_1 and Δ_2 (where we assume that Ω_{U1} and Ω_{U2} are less than π/Δ_1 and π/Δ_2 respectively, the previous expression is written as

$$h(n_1\Delta_1, n_2\Delta_2) = \frac{\Delta_1\Delta_2}{\pi^2}[\Omega_{U2}\,\Omega_{U1}\,\mathrm{sinc}(\Omega_{U2}n_2\Delta_2)\,\mathrm{sinc}(\Omega_{U1}n_1\Delta_1)$$
$$-\Omega_{L2}\,\Omega_{L1}\,\mathrm{sinc}(\Omega_{L2}n_2\Delta_2)\,\mathrm{sinc}(\Omega_{L1}n_1\Delta_1)]$$

Alternative ideal rectangular bandpass filter

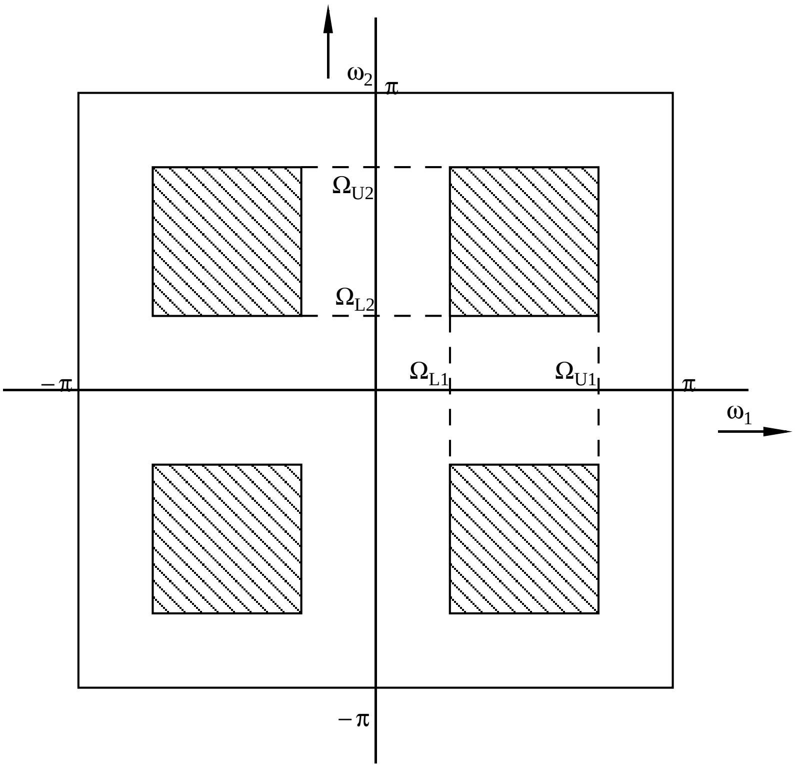


Figure 6: Frequency response of ideal separable rectangular bandpass filter

in this case H is defined by

$$H(\omega_1, \omega_2) = \begin{cases} 1 & \text{if } \Omega_{L1} < |\omega_1| < \Omega_{U1} \text{ and } \Omega_{L2} < |\omega_2| < \Omega_{U2} \\ 0 & \text{otherwise} \end{cases}$$

Note that this definition of the ideal filter is separable:

$$H(\omega_1, \omega_2) = H_1(\omega_1)\, H_2(\omega_2)$$

where $H_i(\omega_i)$ is an ideal 1-D bandpass filter with a lower cut-off frequency of Ω_{Li} and an upper cut-off frequency of Ω_{Ui}, i.e.

$$H_1(\omega_1) = \begin{cases} 1 & \text{if } \Omega_{L1} < |\omega_1| < \Omega_{U1} \\ 0 & \text{otherwise} \end{cases}$$

$$H_2(\omega_2) = \begin{cases} 1 & \text{if } \Omega_{L2} < |\omega_2| < \Omega_{U2} \\ 0 & \text{otherwise} \end{cases}$$

We can therefore write the impulse response of this filter as

$$h(n_1, n_2) = \frac{1}{(2\pi)} \int_{-\pi}^{\pi} H_1(\omega_1)\, e^{j\omega_1 n_1}\, d\omega_1 \frac{1}{(2\pi)} \int_{-\pi}^{\pi} H_2(\omega_2) e^{j\omega_2 n_2}\, d\omega_2 =$$

$$\frac{1}{(2\pi)} \left[\int_{-\Omega_{U1}}^{-\Omega_{L1}} e^{j\omega_1 n_1} d\omega_1 + \int_{\Omega_{L1}}^{\Omega_{U1}} e^{j\omega_1 n_1} d\omega_1 \right] \frac{1}{(2\pi)} \left[\int_{-\Omega_{U2}}^{-\Omega_{L2}} e^{j\omega_2 n_2} d\omega_2 + \int_{\Omega_{L2}}^{\Omega_{U2}} e^{j\omega_2 n_2} d\omega_2 \right]$$

$$= \frac{1}{(2\pi)} \left[\frac{1}{jn_1} \left(e^{-j\Omega_{L1} n_1} - e^{j\Omega_{L1} n_1} \right) - \frac{1}{jn_1} \left(e^{-j\Omega_{U1} n_1} - e^{j\Omega_{U1} n_1} \right) \right]$$
$$\times \frac{1}{(2\pi)} \left[\frac{1}{jn_2} \left(e^{-j\Omega_{L2} n_2} - e^{-j\Omega_{L2} n_2} \right) - \frac{1}{jn_2} \left(e^{-j\Omega_{U2} n_2} - e^{j\Omega_{U2} n_2} \right) \right]$$

$$\boxed{\begin{array}{c} h(n_1, n_2) = \\ \frac{1}{(\pi)} \left[\Omega_{U1} \mathrm{sinc}(\Omega_{U1} n_1) - \Omega_{L1} \mathrm{sinc}(\Omega_{L1} n_1) \right] \frac{1}{(\pi)} \left[\Omega_{U2} \mathrm{sinc}(\Omega_{U2} n_2) - \Omega_{L2} \mathrm{sinc}(\Omega_{L2} n_2) \right] \end{array}}$$

Ideal circularly symmetric bandpass filter

In some cases preferential treatment should not be given to frequency components in any particular direction, $\Longrightarrow$ circularly symmetric frequency responses required:
In this case our frequency response is given by

$$H(\omega_1, \omega_2) = \begin{cases} 1 & \text{if } \Omega_L < |\boldsymbol{\omega}| < \Omega_U \\ 0 & \text{otherwise} \end{cases}$$

where $\boldsymbol{\omega}^2 = \omega_1^2 + \omega_2^2$.

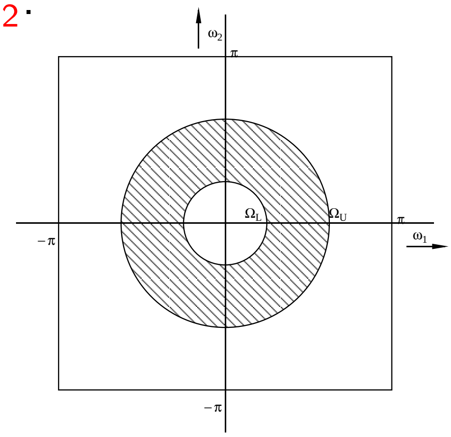


Figure 7: Frequency response of ideal circular bandpass filter

The filter impulse response is then given by:

$$h(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)} \, d\omega_1 \, d\omega_2$$

Transform to polar coordinates r, θ as follows:

$$\omega_1 = r \cos(\theta) \quad \text{and} \quad \omega_2 = r \sin(\theta)$$

$$r^2 = \omega_1^2 + \omega_2^2 \quad , \quad \tan(\theta) = \frac{\omega_2}{\omega_1} \text{ and elemental area} = r \, dr \, d\theta$$

The exponent in the complex exponential of the integrand may be written as:

$$j(\omega_1 \, n_1 + \omega_2 \, n_2) = j \, r \, [n_1 \cos(\theta) + n_2 \sin(\theta)] = j \, r \sqrt{(n_1^2 + n_2^2)} \sin(\theta + \phi)$$

where: $\tan(\phi) = \frac{n_1}{n_2}$

The impulse response may then be written as:

$$h(n_1,n_2) = \frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U}\int_{-\pi}^{\pi} e^{j\,r\sqrt{(n_1^2+n_2^2)}\,\sin(\theta+\phi)}\; r\,d\theta\,dr =$$

$$\frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U} r\int_{-\pi}^{\pi}\left\{\cos\left[r\beta\sin(\theta+\phi)\right]+j\sin\left[r\beta\sin(\theta+\phi)\right]\right\}d\theta dr$$

where $\beta=\sqrt{(n_1^2+n_2^2)}$. A little thought shows that the periodic property of $\sin(\theta)$ ensures that the imaginary part of the expression for $h(n_1,n_2)$ is zero. Thus we must now evaluate the following integral:

$$\frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U} r\int_{-\pi}^{\pi}\left\{\cos\left[r\sqrt{(n_1^2+n_2^2)}\,\sin(\theta+\phi)\right]\right\}\,d\theta\,dr$$

Substitute $\theta' = \theta+\phi$.

$$\frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U} r\int_{\phi-\pi}^{\phi+\pi}\left\{\cos\left[r\sqrt{(n_1^2+n_2^2)}\,\sin(\theta')\right]\right\}\,d\theta'\,dr$$

θ'integrand is clearly a periodic function with period 2π, and since we know that for any function $f(t)$ such that $f(t)=f(t+nT)$, we have

$$\int_0^T f(t)dt = \int_\alpha^{\alpha+T} f(t)dt$$

we can rewrite our integral as

$$\begin{aligned}&\frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U} r\int_{-\pi}^{\pi}\left\{\cos\left[r\sqrt{(n_1^2+n_2^2)}\,\sin(\theta)\right]\right\}\,d\theta\,dr\\ =\;& 2\frac{1}{(2\pi)^2}\int_{\Omega_L}^{\Omega_U} r\int_{0}^{\pi}\left\{\cos\left[r\sqrt{(n_1^2+n_2^2)}\,\sin(\theta)\right]\right\}\,d\theta\,dr\end{aligned}$$

$h(n_1, n_2)$ can now be directly written in terms of the 0th order Bessel function, J_0 where $J_0(x) = \frac{1}{\pi}\int_0^{\pi} \cos\{x \sin(\theta)\}\, d\theta$

$$h(n_1, n_2) = \frac{1}{(2\pi)} \int_{\Omega_L}^{\Omega_U} r\, J_0[r\sqrt{(n_1^2 + n_2^2)}]\, dr$$

Let $r\sqrt{(n_1^2 + n_2^2)} = x$ then:

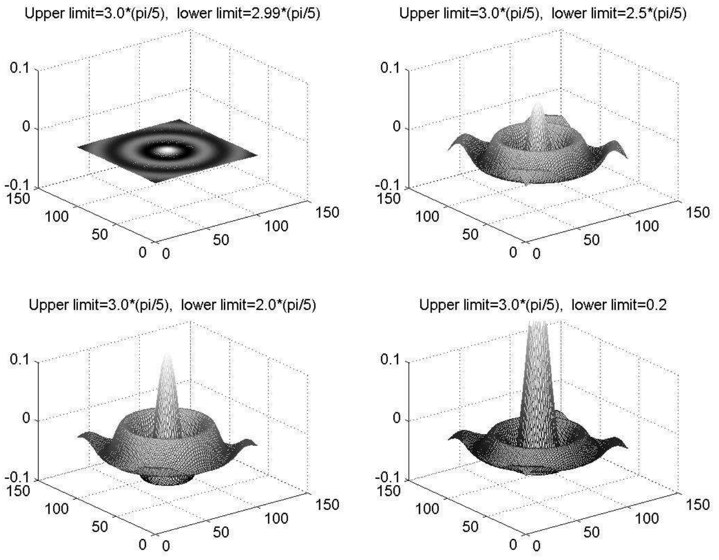
$$h(n_1, n_2) = \frac{1}{2\pi(n_1^2 + n_2^2)} \int_{\Omega_L\sqrt{(n_1^2+n_2^2)}}^{\Omega_U\sqrt{(n_1^2+n_2^2)}} x\, J_0(x)\, dx$$

Now if $J_1(.)$ is the 1st order Bessel function of the first kind where $\int x\, J_0(x)\, dx = x\, J_1(x) \implies$

$$\boxed{h(n_1, n_2) = \frac{1}{2\pi\sqrt{(n_1^2 + n_2^2)}} \{\Omega_U J_1[\sqrt{(n_1^2 + n_2^2)}\Omega_U] - \Omega_L J_1[\sqrt{(n_1^2 + n_2^2)}\Omega_L]\}}$$

Thus, the IR is the difference of two first order Bessel functions. There are many examples of the use of this, one such use is in astronomy.

Impulse response for circularly symmetric bandpass filter



Upper limit=3.0*(pi/5), lower limit=2.99*(pi/5)

Upper limit=3.0*(pi/5), lower limit=2.5*(pi/5)

Upper limit=3.0*(pi/5), lower limit=2.0*(pi/5)

Upper limit=3.0*(pi/5), lower limit=0.2

Figure 8: Inverse FT for four cases of circularly symmetric bandpass filters

IIB 4F8: Image Processing and Image Coding

Handout 4: 2D Digital Filter Design

J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

2-D Digital Filter Design

In designing and using digital filters we firstly specify the characteristics of the filter, secondly we design the filter and thirdly we implement the filter in a discrete setting.

Here we suppose that we are dealing only with real data and that we therefore require a *real* impulse response, $h(n_1, n_2)$ so that the processed image is also real. We will also demand that the impulse response is bounded, implying stability, i.e.

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty$$

An unbounded output will cause many difficulties in practice, possible system overload being one of them.

Have seen that the impulse responses of linear systems can be either finite or infinite $\Longrightarrow$ *finite impulse response* (FIR) or *infinite impulse response* (IIR) filters. Since we require the boundedness condition, we will restrict our attention to FIR filters. Have also seen that a zero-phase 2d filter $(H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2))$ does not disturb the phase characteristics of the image; important for visual perception. $\Longrightarrow$ limit our discussion to *zero-phase FIR filters with real impulse response*.

In general a filter obtained by inverse Fourier transforming some desired zero-phase 2-D frequency response will not have a finite support. However a number of different techniques are available for the design of 2-D digital filters in which the support *is* finite. We will just look at one technique here, the *windowing method* (another common technique is the *frequency transformation method* – discussions of this can be found in the listed textbooks).

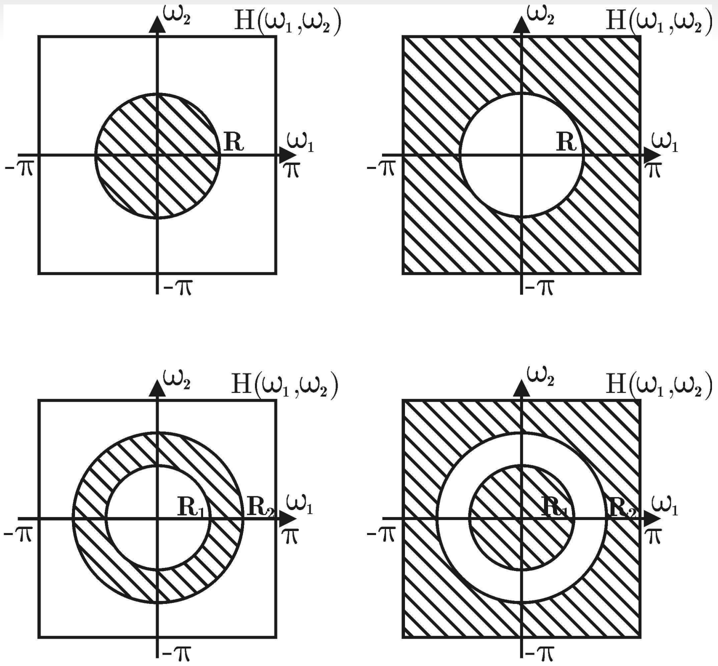


Figure 1: Frequency responses of circularly symmetric ideal filters: lowpass, highpass, bandpass and bandstop

Ideal circularly symmetric filters are shown in figure 1. Ideally, these regions are well delineated, i.e. a lowpass filter only has *passband* and *stopband* regions.

The Window Method

Assume that the desired frequency response $H_d(\omega_1, \omega_2)$ is given – have seen that the desired impulse response $h_d(n_1, n_2)$ can be obtained by taking the inverse FT of H_d.

Idea of the window design method is to multiply the infinite support filter by a window function $w(n_1, n_2)$ which forces the impulse response coefficients $h(n_1, n_2)$ to zero for $n_1, n_2 \notin R_h$ where R_h is the desired support region.

$$h_d(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H_d(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)} \, d\omega_1 \, d\omega_2$$

The filter with finite support is given by:

$$h(n_1, n_2) = h_d(n_1, n_2) \, w(n_1, n_2)$$

Have seen that if $h_d(n_1, n_2)$ and $w(n_1, n_2)$ satisfy the zero-phase requirement, i.e.

$$h_d(-n_1, -n_2) = h_d(n_1, n_2) \;\text{ and }\; w(-n_1, -n_2) = w(n_1, n_2)$$

then the resulting filter will be zero-phase, since the above imply that $h(-n_1, -n_2) = h(n_1, n_2)$.
The frequency response $H(\omega_1, \omega_2)$ of the resulting filter can be expressed in terms of the desired frequency response $H_d(\omega_1, \omega_2)$ and the Fourier transform $W(\omega_1, \omega_2)$ of the window function $w(n_1, n_2)$ as follows.

$$H(\omega_1, \omega_2) = \sum\sum_{(n_1, n_2 \in R_h)} h(n_1, n_2) \, e^{-j(\omega_1 n_1 + \omega_2 n_2)}$$

$$= \sum_{n_2=-\infty}^{\infty} \sum_{n_1=-\infty}^{\infty} w(n_1, n_2) \, h_d(n_1, n_2) \, e^{-j(\omega_1 n_1 + \omega_2 n_2)}$$

Substituting for $h_d(n_1, n_2)$ gives:

$$H(\omega_1, \omega_2) =$$

$$\sum_{n_1, n_2 = -\infty}^{\infty} w(n_1, n_2) \left[\frac{1}{(2\pi)^2} \iint_{-\pi}^{\pi} H_d(\Omega_1, \Omega_2) e^{j(\Omega_1 n_1 + \Omega_2 n_2)} d\Omega_1 d\Omega_2 \right] e^{-j(\omega_1 n_1 + \omega_2 n_2)}$$

$$= \frac{1}{(2\pi)^2} \iint_{-\pi}^{\pi} H_d(\Omega_1, \Omega_2) \sum_{n_1, n_2 = -\infty}^{\infty} w(n_1, n_2) e^{j[(\omega_1 - \Omega_1) n_1 + (\omega_2 - \Omega_2) n_2]} d\Omega_1 d\Omega_2$$

But the summation term is simply the 2-D FT of the window function evaluated at frequencies $\omega_1 - \Omega_1, \omega_2 - \Omega_2$.

$$\therefore \boxed{\begin{array}{l} H(\omega_1, \omega_2) = \dfrac{1}{(2\pi)^2} \displaystyle\iint_{-\pi}^{\pi} H_d(\Omega_1, \Omega_2) W((\omega_1 - \Omega_1), (\omega_2 - \Omega_2)) \, d\Omega_1 \, d\Omega_2 \\ W(\omega_1, \omega_2) = \displaystyle\sum_{n_2 = -\infty}^{\infty} \sum_{n_1 = -\infty}^{\infty} w(n_1, n_2) \, e^{-j(\omega_1 n_1 + \omega_2 n_2)} \end{array}}$$

Thus the actual filter frequency response $H(\omega_1, \omega_2)$ is given by the **convolution** of the desired frequency response $H_d(\omega_1, \omega_2)$ with the window function spectrum $W(\omega_1, \omega_2)$.

This is exactly as we should expect since we multiply in the spatial domain and must therefore convolve in the frequency domain.

Thus the effect of the window is to smooth H_d – clearly we would prefer to have the mainlobe width of $W(\omega_1, \omega_2)$ small so that H_d is changed as little as possible. We also want sidebands of small amplitude so that the ripples in the (ω_1, ω_2) plane outside the region of interest are kept small.

Next – look at methods of producing window functions.

Product of 1-D Windows

One popular method for obtaining a 2-D window is to simply take the product of two 1-D windows:

$$w(n_1, n_2) = w_1(n_1)\ w_2(n_2)$$

If the above holds, it is not hard to see that the FT of W is also separable, i.e.

$$W(\omega_1, \omega_2) = W_1(\omega_1)\ W_2(\omega_2)$$

Note also that if the desired frequency response $H_d(\omega_1, \omega_2)$ is separable then the desired impulse response $h_d(n_1, n_2)$ is also separable (clear from previous equations).
Thus the windowed impulse response is also separable and the whole design process may reduce essentially to that of designing two 1-D filters.

Rectangular Window

We now investigate the spectra of a number of window functions formed from the product of 1-D windows.

First consider a rectangular window function given simply by

$$w(u_1, u_2) = \begin{cases} 1 & \text{if } |u_1| < U_1 \text{ and } |u_2| < U_2 \\ 0 & \text{otherwise} \end{cases}$$

or, since w is separable we have $w = w_1 w_2$ where

$$w_i(u_i) = \begin{cases} 1 & \text{if } |u_i| < U_i \\ 0 & \text{otherwise} \end{cases}$$

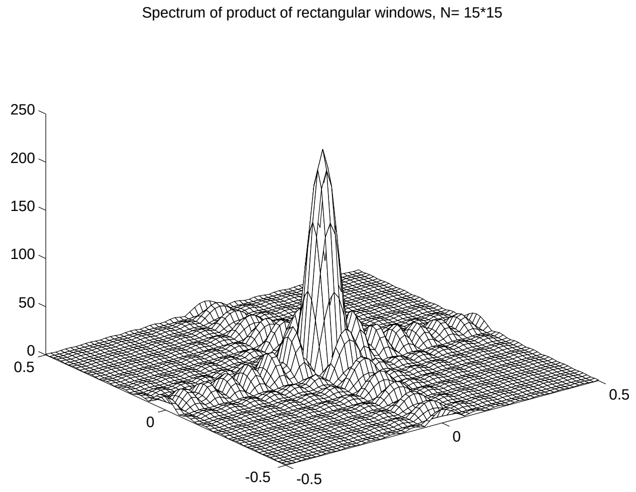


Figure 2: Spectrum of product of rectangular windows, support = 15×15

Figure 2 is formed by taking the inverse FT of a zero-padded version of the array/image $w(n_1, n_2)$.

Cosine Window

Consider a 1-d cosine function window,

$$w(u_1) = \begin{cases} cos(\pi \frac{u_1}{U_1}) & \text{if } |u_1| < U_1 \\ 0 & \text{otherwise} \end{cases}$$

Taking the FT of this can be shown to give

$$W_1(\omega_1) = U_1\{\text{sinc}(\pi - \omega_1 U_1) + \text{sinc}(\pi + \omega_1 U_1)\}$$

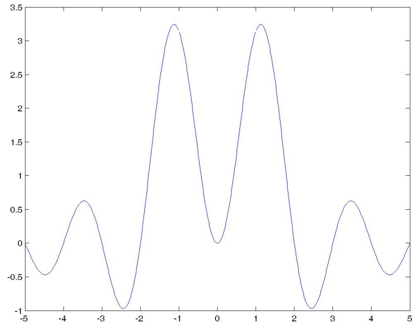


Figure 3: Spectrum of a 1d cosine window

This is hardly ideal!

Hamming Window

The 1d Hamming window function is simply

$$w(u_1) = \begin{cases} 0.54 + 0.46\cos(\pi \frac{u_1}{U_1}) & \text{if } |u_1| < U_1 \\ 0 & \text{otherwise} \end{cases}$$

Note, that when $u_1 = 0$, $w = 1$, when $u_1 = U_1/2$, $w = 0.54$ and when $u_1 = U_1$, $w = 0.54 - 0.46 = 0.08$. Thus the 1d Hamming window decays relatively slowly down to a value of 0.08 at $u_1 = U_1$. The 2d Hamming window is therefore defined as

$$w(u_1, u_2) = \begin{cases} \left[0.54 + 0.46\cos(\pi \frac{u_1}{U_1})\right] \left[0.54 + 0.46\cos(\pi \frac{u_2}{U_2})\right] & \text{if } |u_1| < U_1 \ \ |u_2| < U_2 \\ 0 & \text{otherwise} \end{cases}$$

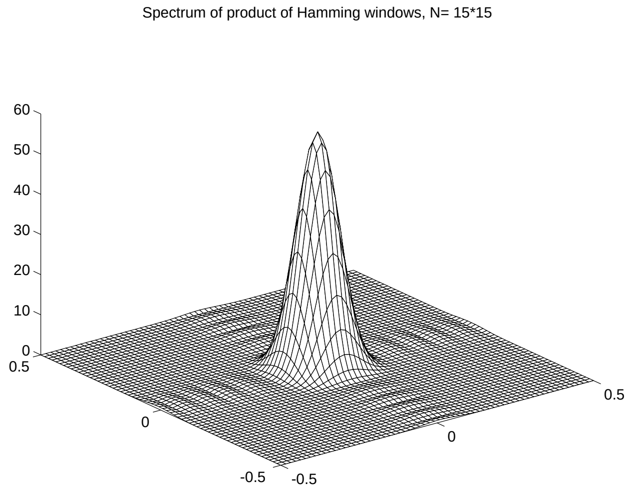


Figure 4: Spectrum of product of Hamming windows,support $= 15 \times 15$

Again, Figure 4 is formed by taking the FT of a zero-padded version of array/image $w(n_1, n_2)$ given on previous slide.

Kaiser Window

The 1d Kaiser window function is given by

$$w(u_1) = \begin{cases} \frac{I_o(\beta\sqrt{(1-(\frac{u_1}{U_1})^2)}}{I_o(\beta)} & \text{if } |u_1| < U_1 \\ 0 & \text{otherwise} \end{cases}$$

Here $I_0(.)$ is the modified Bessel function of the 1st kind of order zero. The parameter β allows the spectral mainlobe performance to be traded against spectral sidelobe performance.

Increasing β widens the mainlobe and increases the attenuation of the sidelobes.

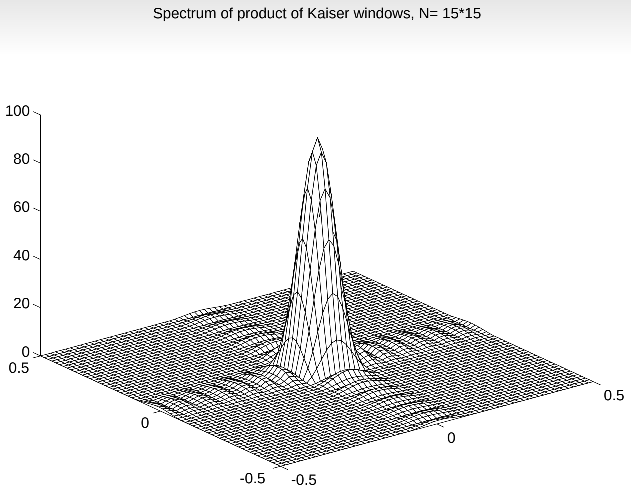


Figure 5: Spectrum of product of Kaiser windows, $\beta = 3$, support = 15×15

We see that the rectangular window has a good mainlobe behaviour (small width) but a poor sidelobe behaviour (large peaks); the Hamming window has a moderate mainlobe behaviour but fairly good sidelobe behaviour; for the Kaiser window the relative behaviour of the main and side-lobes can be controlled by the parameter β.

Windowing ideal circularly symmetric filters

Now consider ideal circularly symmetric lowpass filters – use the bandpass filters we have derived previously with $\Omega_L = 0$. We have seen that the desired impulse response, $h_d(n_1, n_2)$ of such a filter is given by

$$h(n_1, n_2) = \frac{1}{2\pi\sqrt{(n_1^2 + n_2^2)}} \{\Omega_U J_1[\sqrt{(n_1^2 + n_2^2)}\Omega_U]\}$$

We now apply the separable 2d window functions to this impulse response – again, use rectangular, Hamming and Kaiser. To find the new impulse response, we multiply by the window and to find the new frequency response we convolve H_d with the FT of the window.

Rectangular window

The impulse and frequency responses of an ideal circular low-pass filter, with a normalised cut-off frequency of $\Omega = 0.25$, designed using a rectangular window are shown below

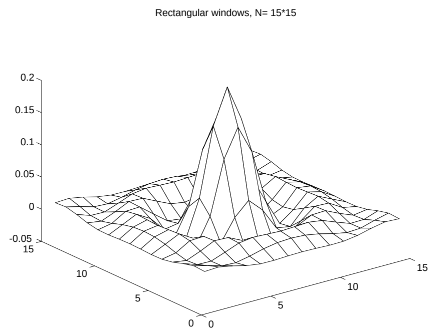


Figure 6: LP filter impulse response, rectangular window product, support = 15×15

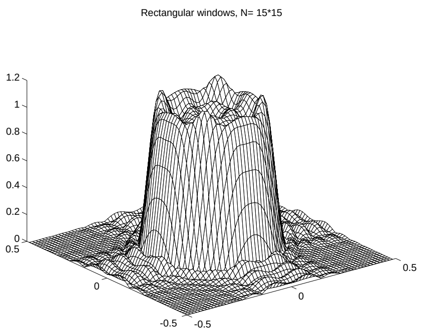


Figure 7: LP filter frequency response, rectangular window product, support = 15×15

Hamming window

The impulse and frequency responses of an ideal circular low-pass filter, with a normalised cut-off frequency of $\Omega = 0.25$, designed using a Hamming window are shown below

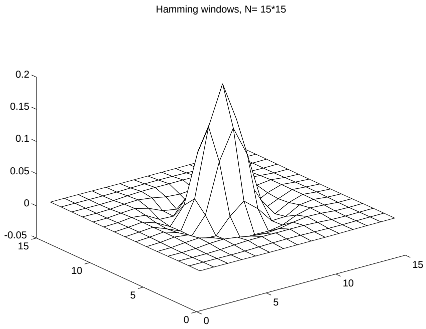


Figure 8: LP filter impulse response, Hamming window product, support = 15×15

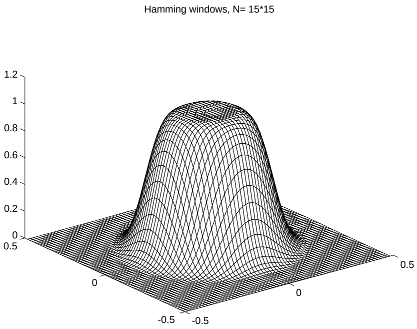


Figure 9: LP filter frequency response, Hamming window product, support = 15×15

Kaiser window

The impulse and frequency responses of an ideal circular low-pass filter, with a normalised cut-off frequency of $\Omega = 0.25$, designed using a Kaiser window are shown below

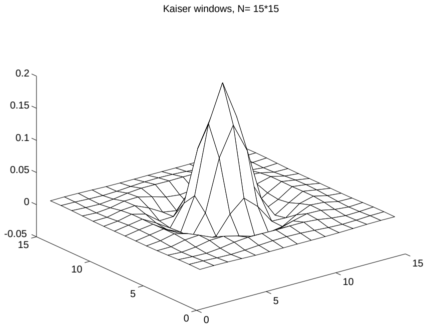


Figure 10: LP filter impulse response, Kaiser window ($\beta = 3$), support = 15×15

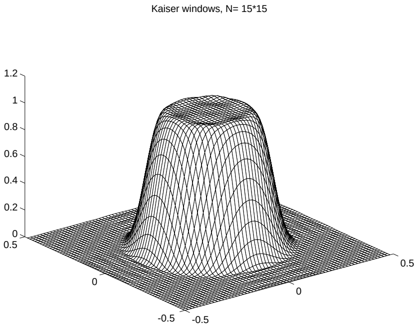


Figure 11: LP filter frequency response, Kaiser ($\beta = 3$) window, support = 15×15

Rotation of 1-D Windows

A second approach to implementing the window design method is to obtain a 2-D *continuous* window function $w(u_1, u_2)$ by rotating a 1-D continuous window $w_1(u)$.

$$\boxed{w(u_1, u_2) = w_1(u)|_{u=\sqrt{(u_1^2+u_2^2)}}}$$

The continuous 2-D window is then sampled to produce a discrete 2-D window $w(n_1, n_2)$:

$$\boxed{w(n_1, n_2) = w(u_1, u_2)|_{u_1=n_1\,\Delta_1,\; u_2=n_2\,\Delta_2}}$$

The spectrum $W_s(\omega_1, \omega_2)$ of the 2-D discrete window is given by: (see derivation of 2-D sampling theorem)

$$W_s(\omega_1, \omega_2) = \frac{1}{\Delta_1 \Delta_2} \sum_{p_2=-\infty}^{\infty} \sum_{p_1=-\infty}^{\infty} W(\omega_1 - p_1\Omega_1, \omega_2 - p_2\Omega_2)$$

where:

$$\Omega_1 = \frac{2\pi}{\Delta_1}, \qquad \Omega_2 = \frac{2\pi}{\Delta_2}$$

and $W(\omega_1, \omega_2)$ is the Fourier transform of the continuous 2-D window $w(u_1, u_2)$. Note that W is in general not equivalent to the rotated form of the 1d FT.

Note that even if $W(\omega_1, \omega_2)$ has circular symmetry the resulting $W_s(\omega_1, \omega_2)$ will not necessarily be symmetric as a result of possible aliasing.

Following figures show the spectra of a number of window functions formed by rotating 1-D window functions.

Spectrum of rotated rectangular window

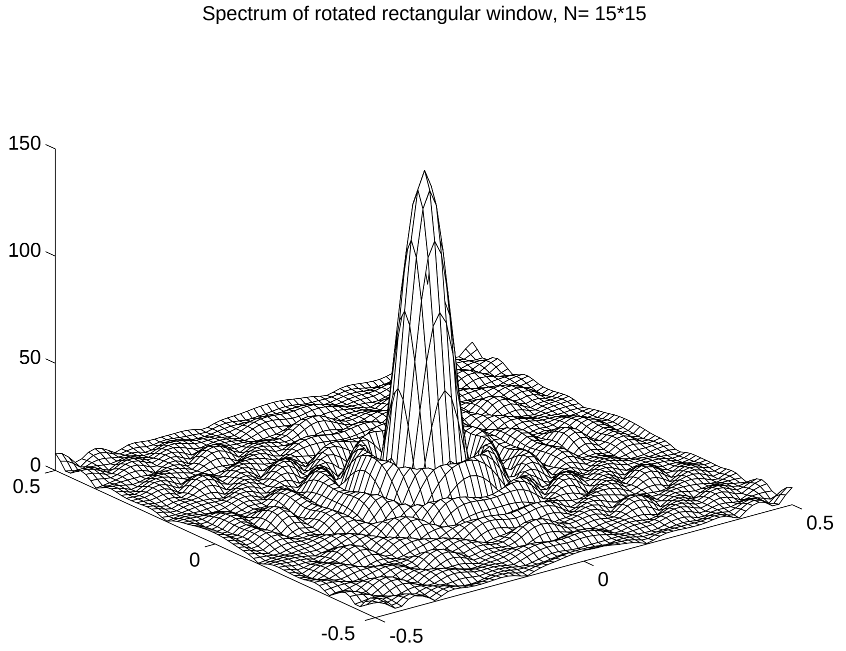


Figure 12: Spectrum of rotated rectangular window, support = 15×15

Spectrum of rotated Hamming window

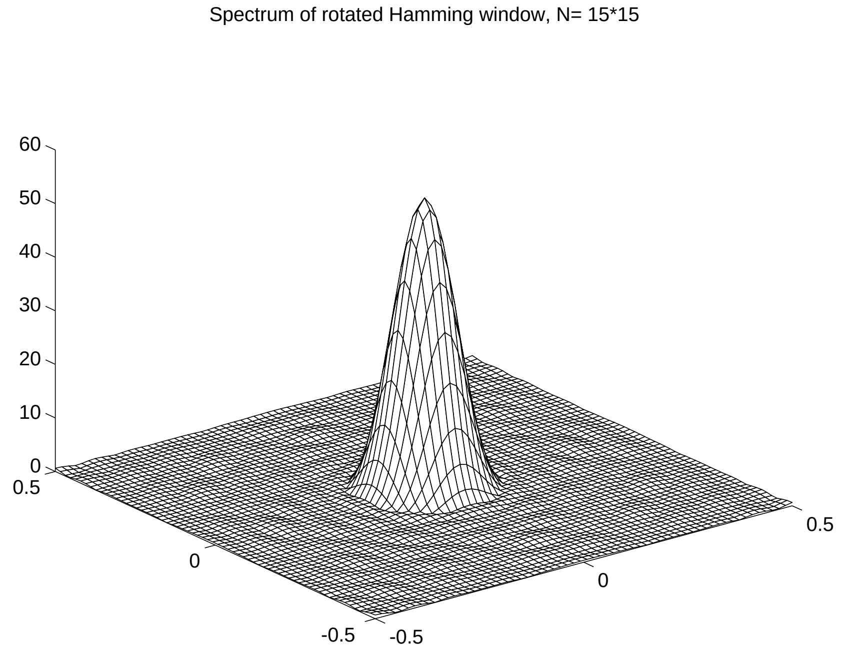


Figure 13: Spectrum of rotated Hamming window, support = 15×15

Spectrum of rotated Kaiser window

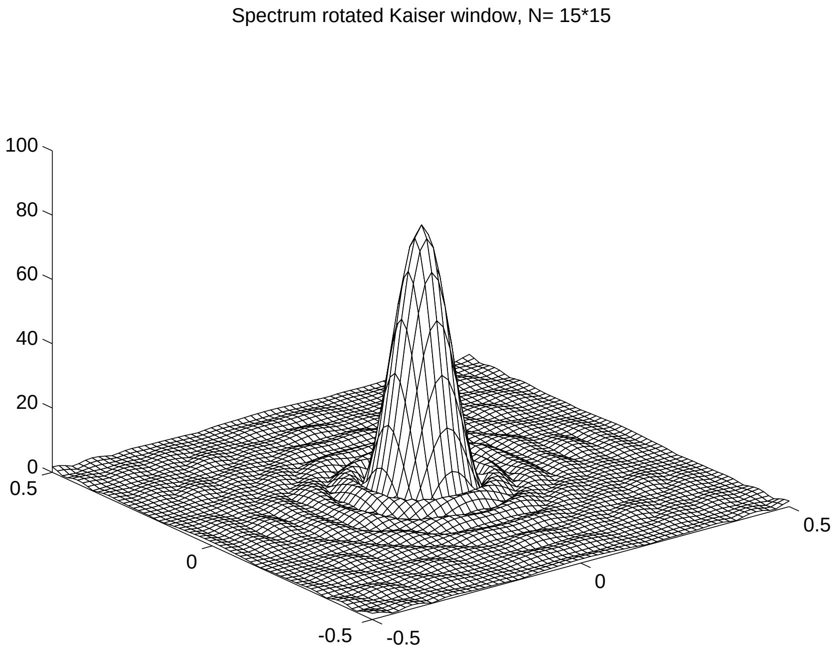


Figure 14: Spectrum of rotated Kaiser window ($\beta = 3$), support = 15×15

Impulse and Frequency response of windowed circular LP filters

As we did with the separable windows, we can now look at the impulse and frequency responses of circular low-pass filters, with a normalised cut-off frequency of $\Omega = 0.25$, designed using the various rotated windows.

Once again to obtain the impulse response we simply multiply the ideal impulse response with the window and to obtain the frequency response, we convolve the ideal frequency response with the FT of the window. The derived frequency responses for the three types of windowing functions we are considering here are shown in the following figures.

h and H of a windowed LP filter – rectangular window

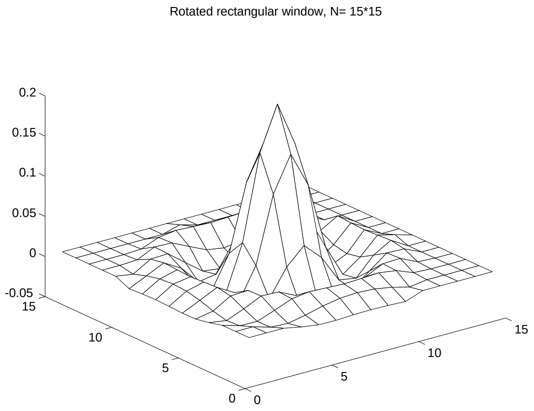


Figure 15: LP filter impulse response, rotated rectangular window, support = 15×15

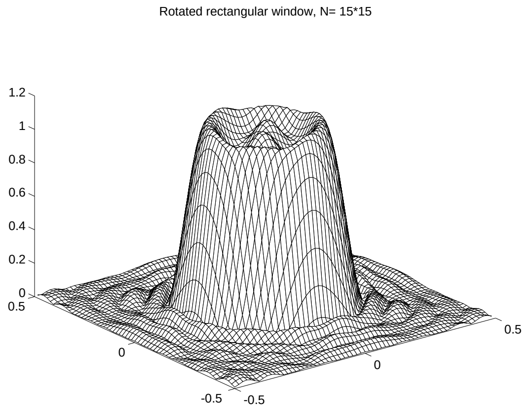


Figure 16: LP filter frequency response, rotated rectangular window, support = 15×15

h and H of a windowed LP filter – Hamming window

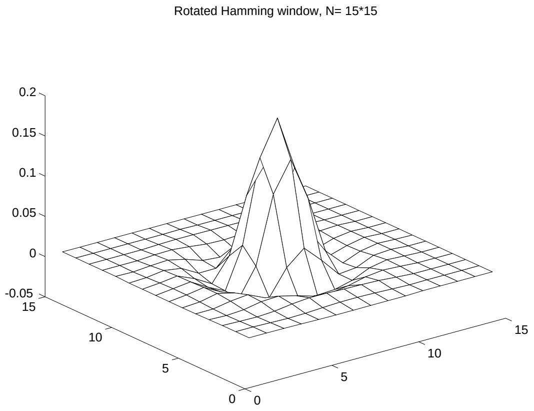


Figure 17: LP filter impulse response, rotated Hamming window, support = 15×15

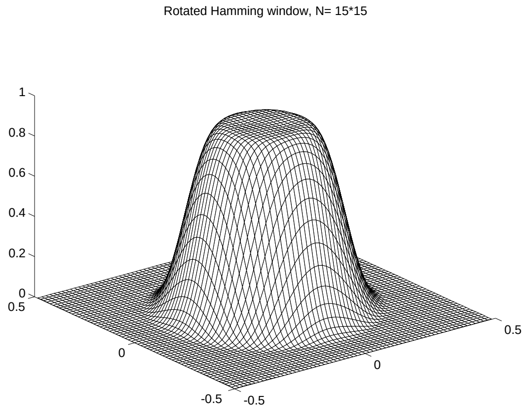


Figure 18: LP filter frequency response, rotated Hamming window, support = 15×15

h and H of a windowed LP filter – Kaiser window

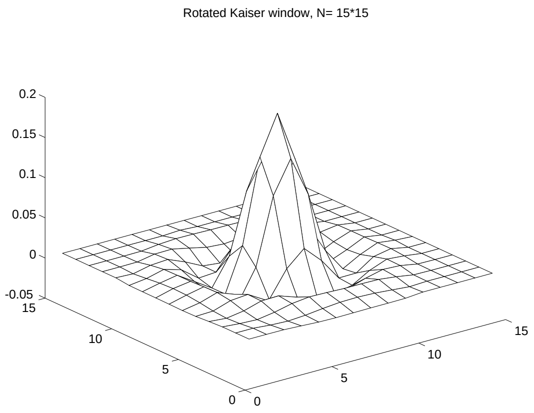


Figure 19: LP filter impulse response, rotated Kaiser window, support = 15×15

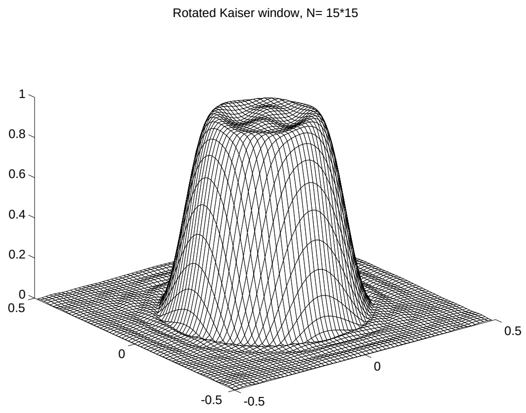


Figure 20: LP filter frequency response, rotated Kaiser window, support = 15×15

IIB 4F8: Image Processing and Image Coding

Handout 5 : Image Deconvolution

J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

Introduction to Deconvolution

We are often concerned with estimating an image x from data y measured with an imperfect instrumentation system. Typically, *blurring* is introduced as a result of an imperfect optical system and noise d as a result of an imperfect imaging device.
The observed image $y(u_1, u_2)$ may be expressed as:

$$y(u_1, u_2) = f\left[x(u_1, u_2)\right] + d(u_1, u_2)$$

where $f(.)$ is, in general, a nonlinear function of the original image $x(u_1, u_2)$. However we first consider only the case of linear distortion.

$$y(u_1, u_2) = L\left[x(u_1, u_2)\right] + d(u_1, u_2)$$

where the linear operator is a 2-dimensional convolution:

$$L\left[x(u_1,u_2)\right] = \int\int h(v_1,v_2)x(u_1-v_1,u_2-v_2)dv_1dv_2$$

where h is the *impulse response or point spread function*.
Assume a discrete space model:

$$y(n_1,n_2) = L\left[x(n_1,n_2)\right] + d(n_1,n_2)$$

where:

$$L\left[x(n_1,n_2)\right] = \sum_{m_2=-\infty}^{\infty}\sum_{m_1=-\infty}^{\infty} h(m_1,m_2)x(n_1-m_1,n_2-m_2)$$

Consider first the case where there is no noise and the objective is to recover the image $x(n_1,n_2)$ from the observations. Assume that the impulse response of the distorting system is known.

Deconvolution of Noiseless Images

Problem: given observations $y(n_1,n_2)$ and knowledge of the function $h(m_1,m_2)$ obtain an estimate of the original image $x(u_1,u_2)$ where:

$$y(n_1,n_2) = \sum_{m_1}\sum_{m_2} h(m_1,m_2)x(n_1-m_1,n_2-m_2)$$

x and y are convolved $\Longrightarrow$ take the Fourier transform of each side of the above to give:

$$Y(\omega_1,\omega_2) = H(\omega_1,\omega_2)X(\omega_1,\omega_2)$$

where: $H(\omega_1,\omega_2) = \sum_{n_2=-\infty}^{\infty}\sum_{n_1=-\infty}^{\infty} h(n_1,n_2)e^{-j(\omega_1 n_1+\omega_2 n_2)}$

$$\therefore X(\omega_1,\omega_2) = \frac{Y(\omega_1,\omega_2)}{H(\omega_1,\omega_2)}$$

$$\Longrightarrow \quad x(n_1,n_2) = \frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} X(\omega_1,\omega_2)e^{j(\omega_1 n_1+\omega_2 n_2)}d\omega_1 d\omega_2$$

Note: If $H(\omega_1, \omega_2)$ has zeros, then the inverse filter, $1/H$, will have infinite gain. i.e. if $1/H$ is very large (or indeed infinite), small noise in the regions of the frequency plane where these large values of $1/H$ occur can be hugely amplified. To counter this we can threshold the frequency response, leading to the so-called, pseudo-inverse or generalised inverse filter $H_g(\omega_1, \omega_2)$ given by

$$H_g(\omega_1, \omega_2) = \begin{cases} \frac{1}{H(\omega_1,\omega_2)} & \frac{1}{|H(\omega_1,\omega_2|} < \gamma \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

or

$$H_g(\omega_1, \omega_2) = \begin{cases} \frac{1}{H(\omega_1,\omega_2)} & \frac{1}{|H(\omega_1,\omega_2|} < \gamma \\ \gamma \frac{|H(\omega_1,\omega_2|}{H(\omega_1,\omega_2)} & \text{otherwise} \end{cases} \tag{2}$$

Clearly for $\frac{1}{|H(\omega_1,\omega_2|} \geq \gamma$ in equation 2, the modulus of the filter is set as γ, whereas in previous equation it is set as 0.

Although the pseudo-inverse filter may perform reasonably well on noiseless images, the performance is unsatisfactory with even mildly noisy images due to the still significant noise gain at frequencies where $H(\omega_1, \omega_2)$ is relatively small; $\Longrightarrow$ investigate the possibility of taking the additive noise into account in a more formal manner.

Now look at two examples of inverse filtering.

First use the inverse filtering method on an image which has been convolved with a gaussian (assuming knowledge of the blurring gaussian).

Then use the same inverse filtering operation when the original image has been convolved with the same gaussian and then subjected to additive noise.

Results of Inverse Filterng

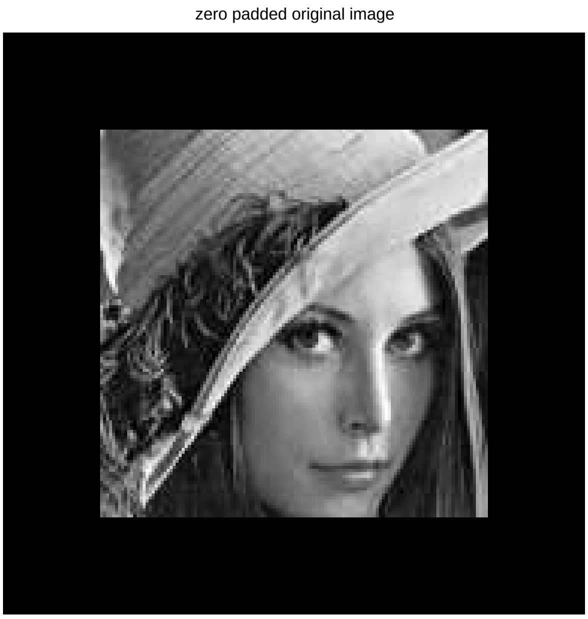


Figure 1: 192×192 original image; central 128×128 of Lenna padded with zeros

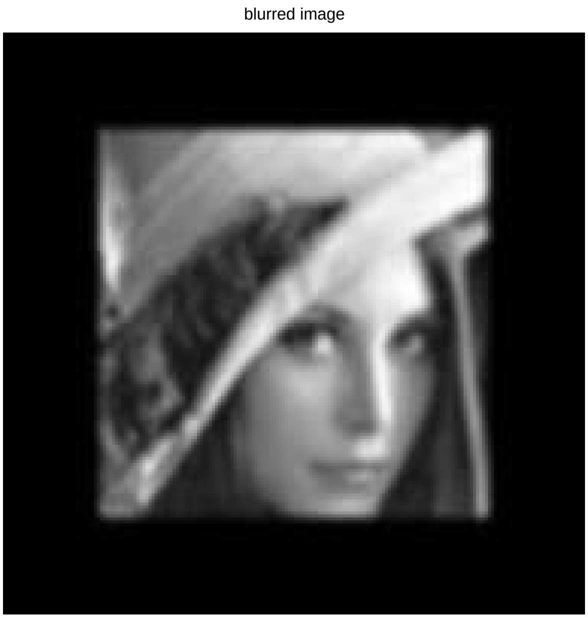


Figure 2: Previous image blurred with a gaussian

Results of Inverse Filtering

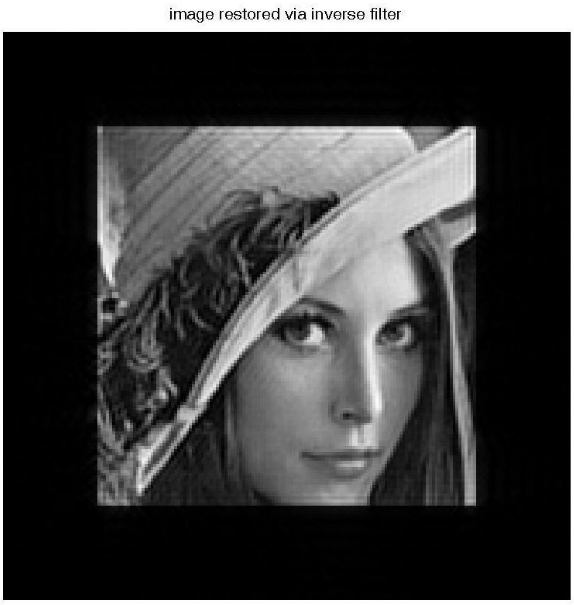


Figure 3: Inverse filter applied to noiseless blurred image

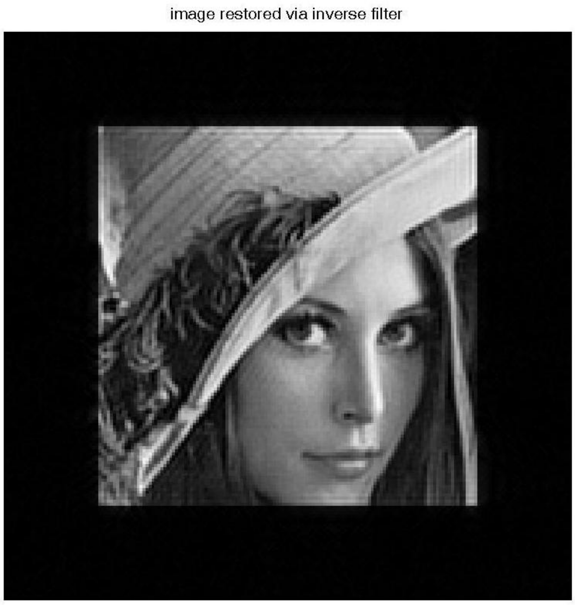


Figure 4: Inverse filter applied to blurred image plus noise (0.01)

Results of Inverse Filtering

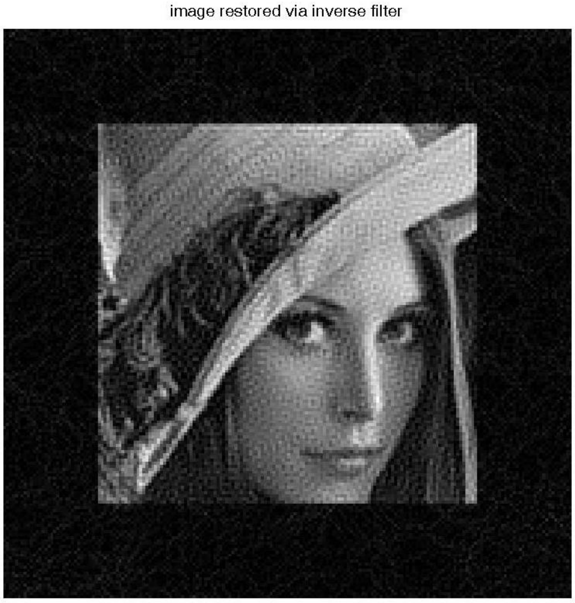


Figure 5: Inverse filter applied to blurred image plus noise (0.1)

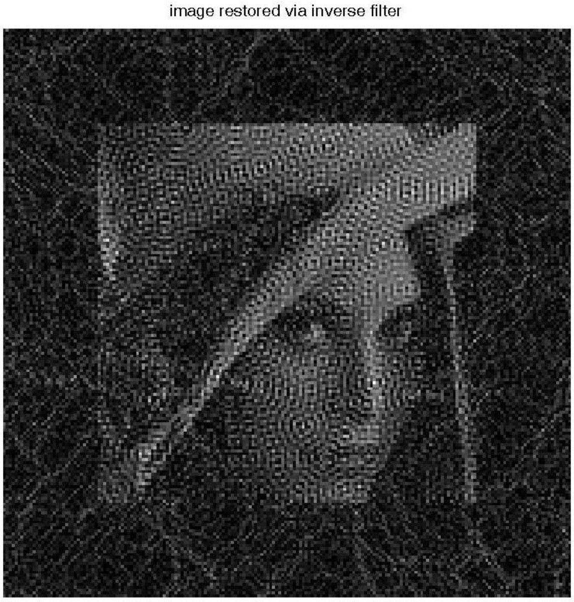


Figure 6: Inverse filter applied to blurred image plus noise (0.5)

Note: in giving the above examples of the inverse filter in action we used a zero-padded image. We do this because if we don't, the convolution operator (depending on the particular algorithm we are using) may convolve pixels near the edge of the image with other parts of the image (via the periodic tiling of the plane).

The easiest way of avoiding this is to put a guard band of zeros (the width of the guard band will depend upon the width of the convolving function) around the image.

The whole of the theory of 2d Fourier Transforms requires periodic boundary conditions (ie repeating the image to form a tiling of the plane); when we convolve with a blurring function, this ruins the periodic boundary conditions unless there is an adequate edge padding of zeros.

Notation

Mathematical development of optimal image processing algorithms requires a significant amount of manipulation of convolution expressions:

$$y(n_1, n_2) = \sum_{m_2=-\infty}^{\infty} \sum_{m_1=-\infty}^{\infty} h(m_1, m_2)\, x(n_1 - m_1, n_2 - m_2) \qquad (3)$$

This may be rewritten more compactly using vectors:

$$\mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad \text{and} \quad \mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$$

The range of the convolution limits can be expressed in terms of the set $\mathbf{Z} = \{n : -\infty \leq n \leq \infty,\ n = \text{integer}\} \implies$ the convolution equation, 3, becomes:

$$y(\mathbf{n}) = \sum_{\mathbf{m} \in \mathbf{Z}^2} h(\mathbf{m})\, x(\mathbf{n} - \mathbf{m})$$

We will of course be constantly using Fourier transforms:

$$H(\omega_1, \omega_2) = \sum_{n_2=-\infty}^{\infty} \sum_{n_1=-\infty}^{\infty} h(n_1, n_2)\, e^{-j(\omega_1 n_1 + \omega_2 n_2)}$$

$$h(n_1, n_2) = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)}\, d\omega_1\, d\omega_2$$

which may be expressed in vector notation as:

$$H(\boldsymbol{\omega}) = \sum_{\mathbf{n} \in \mathbf{Z}^2} h(\mathbf{n})\, e^{-j\boldsymbol{\omega}^T \mathbf{n}}$$

$$h(\mathbf{n}) = \frac{1}{(2\pi)^2} \int_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} H(\boldsymbol{\omega})\, e^{j\boldsymbol{\omega}^T \mathbf{n}}\, \mathbf{d}\boldsymbol{\omega}$$

$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$ and $\boldsymbol{\Omega} = \{\omega_1, \omega_2 :\ -\pi \leq \omega_1 \leq \pi,\ -\pi \leq \omega_2 \leq \pi\}$

Note that notation is not limited to 2-D analysis. (Above is assuming normalised frequencies).

Some revision

Let us regard our arrays of (possibly complex) numbers $x(n_1, n_2)$ (or $x(\mathbf{n})$) and $y(n_1, n_2)$ (or $y(\mathbf{n})$) as random variables. The *cross correlation* function of x and y is defined as R_{xy} where

$$R_{xy}(a_1, a_2; b_1, b_2) = E[x(a_1, a_2)y^*(b_1, b_2)]$$

or

$$R_{xy}(\mathbf{a}, \mathbf{b}) = E[x(\mathbf{a})y^*(\mathbf{b})]$$

where $E[.]$ is the expectation and $*$ denotes complex conjugate. The *autocorrelation* function of x is R_{xx} where

$$R_{xx}(\mathbf{a}, \mathbf{b}) = E[x(\mathbf{a})x^*(\mathbf{b})]$$

For *spatially stationary processes* $x(\mathbf{n})$, $y(\mathbf{n})$, the correlation function is ‘translationally invariant’, i.e.

$$R_{xy}(\mathbf{0}, \mathbf{n}) \equiv R_{xy}(\mathbf{n}) = E[x(\mathbf{k})y^*(\mathbf{k} - \mathbf{n})] \;\; \forall \mathbf{k}$$

i.e. the cross-correlation between the origin in the x image and the point $\mathbf{n}$ in the y image is independent of where the origin is taken. The *cross-power spectrum* of two jointly stationary processes $x(\mathbf{n})$ and $y(\mathbf{n})$ is written as $P_{xy}(\boldsymbol{\omega})$ and is given by the FT of the cross-correlation function

$$P_{xy}(\boldsymbol{\omega}) = FT(R_{xy}(\mathbf{n}))$$

Note here that we could have chosen the following convention for the cross correlation (denote as R_{xy}^{+})

$$R_{xy}^{+}(\mathbf{n}) \equiv R_{xy}(-\mathbf{n}) = E[x(\mathbf{k})y^*(\mathbf{k} + \mathbf{n})] \;\; \forall \mathbf{k}$$

Let us look at how the two forms are related.

$$R_{xy}(-\mathbf{n}) = \{E[y(\mathbf{k}+\mathbf{n})x^*(\mathbf{k})]\}^* = \{E[y(\mathbf{p})x^*(\mathbf{p}-\mathbf{n})]\}^* = R^*_{yx}(\mathbf{n})$$

Thus we see that $R^+_{xy}(\mathbf{n}) = R_{xy}(-\mathbf{n}) = R^*_{yx}(\mathbf{n})$. For the autocorrelation this becomes

$$R_{xx}(-\mathbf{n}) = R^*_{xx}(\mathbf{n})$$

Since the power spectrum is the FT of the autocorrelation we see that

$$P^*_{xx}(\boldsymbol{\omega}) = \sum_{\mathbf{p}} R^*_{xx}(\mathbf{p})e^{j\boldsymbol{\omega}^T\mathbf{p}} = \sum_{\mathbf{p}} R_{xx}(-\mathbf{p})e^{j\boldsymbol{\omega}^T\mathbf{p}}$$

so that

$$P^*_{xx}(\boldsymbol{\omega}) = \sum_{\mathbf{q}} R_{xx}(\mathbf{q})e^{-j\boldsymbol{\omega}^T\mathbf{q}} = P_{xx}(\boldsymbol{\omega})$$

which indicates that $P_{xx}(\boldsymbol{\omega})$ is therefore real.

The Wiener Filter

Using the vector notation, the image model:

$$y(n_1,n_2) = \sum_{m_2=-\infty}^{\infty}\sum_{m_1=-\infty}^{\infty} h(m_1,m_2)x(n_1-m_1,n_2-m_2) + d(n_1,n_2)$$

may be rewritten as:

$$y(\mathbf{n}) = \sum_{\mathbf{m}\in\mathbf{Z}^2} h(\mathbf{m})\,x(\mathbf{n}-\mathbf{m}) + d(\mathbf{n})$$

Assume that an estimate $\hat{x}(\mathbf{n})$ of the image $x(\mathbf{n})$ is to be obtained by linear spatially-invariant filtering of the observed image $y(\mathbf{n})$.

$$\hat{x}(\mathbf{n}) = \sum_{\mathbf{q}\in\mathbf{Z}^2} g(\mathbf{q})\,y(\mathbf{n}-\mathbf{q})$$

It is now required to choose the filter impulse response $g(\mathbf{q})$ such that the expectation of the squared error is minimised.

$$\text{Minimise } Q = E\{[x(\mathbf{n}) - \hat{x}(\mathbf{n})]^2\}$$

$$= E\left\{\left[x(\mathbf{n}) - \sum_{\mathbf{q}\in\mathbf{Z^2}} g(\mathbf{q})\, y(\mathbf{n}-\mathbf{q})\right]^2\right\}$$

Differentiate the objective function with respect to $g(\mathbf{p})$

$$\frac{\partial Q}{\partial g(\mathbf{p})} = E\left\{2\left[x(\mathbf{n}) - \sum_{\mathbf{q}\in\mathbf{Z^2}} g(\mathbf{q})\, y(\mathbf{n}-\mathbf{q})\right][-y(\mathbf{n}-\mathbf{p})]\right\} = 0 \quad \forall \mathbf{p} \in \mathbf{Z^2}$$

$$\therefore\; E\{x(\mathbf{n})\, y(\mathbf{n}-\mathbf{p})\} = \sum_{\mathbf{q}} g(\mathbf{q})\, E\{y(\mathbf{n}-\mathbf{q})\, y(\mathbf{n}-\mathbf{p})\} \qquad (4)$$

Assuming that the images are spatially stationary (with x and y real), we can then write:

$$E\{x(\mathbf{n})\, y(\mathbf{n}-\mathbf{p})\} = R_{xy}(\mathbf{p})$$

If we rewrite $y(\mathbf{n}-\mathbf{p})$ as $y(\mathbf{n}-\mathbf{q}+\mathbf{q}-\mathbf{p})$, the expectation in the RHS of equation 4 becomes

$$E\{y(\mathbf{n}-\mathbf{q})\, y(\mathbf{n}-\mathbf{p})\} = E\{y(\mathbf{n}-\mathbf{q})\, y(\mathbf{n}-\mathbf{q}+\mathbf{q}-\mathbf{p})\}$$

$$= E\{y(\mathbf{k})\, y(\mathbf{k}+\mathbf{q}-\mathbf{p})\} = R_{yy}(\mathbf{p}-\mathbf{q})$$

$$\boxed{\therefore\; R_{xy}(\mathbf{p}) = \sum_{\mathbf{q}} g(\mathbf{q})\, R_{yy}(\mathbf{p}-\mathbf{q}) \quad \forall \mathbf{p} \in \mathbf{Z^2}} \qquad (5)$$

These are the *Normal* or *Wiener-Hopf* equations and can, in principle, be solved for filter impulse response $g(\mathbf{q})$. Take the FT of this equation

$$\sum_{\mathbf{p}\in\mathbf{Z}^2} R_{xy}(\mathbf{p})e^{-j\boldsymbol{\omega}^T\mathbf{p}} = \sum_{\mathbf{p}\in\mathbf{Z}^2}\left\{\sum_{\mathbf{q}} g(\mathbf{q})\, R_{yy}(\mathbf{p}-\mathbf{q})\right\} e^{-j\boldsymbol{\omega}^T\mathbf{p}}$$

$$= \sum_{\mathbf{q}} g(\mathbf{q}) \sum_{\mathbf{p}} R_{yy}(\mathbf{p}-\mathbf{q})\, e^{-j\boldsymbol{\omega}^T\mathbf{p}}$$

Let $\mathbf{p}-\mathbf{q}=\mathbf{k}$, and write the FTs of R_{xy} and $g(\mathbf{q})$ as $P_{xy}(\boldsymbol{\omega})$ and $G(\boldsymbol{\omega})$ then:

$$P_{xy}(\boldsymbol{\omega}) = \sum_{\mathbf{q}} g(\mathbf{q}) \sum_{\mathbf{k}} R_{yy}(\mathbf{k})\, e^{-j\boldsymbol{\omega}^T(\mathbf{q}+\mathbf{k})} = \sum_{\mathbf{q}} g(\mathbf{q})\, e^{-j\boldsymbol{\omega}^T\mathbf{q}} \sum_{\mathbf{k}} R_{yy}(\mathbf{k})\, e^{-j\boldsymbol{\omega}^T\mathbf{k}}$$

$$\therefore P_{xy}(\boldsymbol{\omega}) = G(\boldsymbol{\omega})\, P_{yy}(\boldsymbol{\omega}) \quad \Longrightarrow \quad \boxed{G(\boldsymbol{\omega}) = \frac{P_{xy}(\boldsymbol{\omega})}{P_{yy}(\boldsymbol{\omega})}}$$

This is the optimum linear filter for recovering the original image $x(\mathbf{n})$ from the noisy measurements $y(\mathbf{n})$. Note that we must estimate P_{xy} from the data.

Uncorrelated observation noise

Consider W-H equations when the observation noise $d(\mathbf{n})$ is uncorrelated with the image.

$$R_{yy}(\mathbf{p}) = E\{y(\mathbf{n})\, y(\mathbf{n}-\mathbf{p})\} \text{ where } y(\mathbf{n}) = \sum_{\mathbf{m}} h(\mathbf{m})\, x(\mathbf{n}-\mathbf{m}) + d(\mathbf{n})$$

If signal and noise are uncorrelated and noise is zero mean:

$$R_{yy}(\mathbf{p}) = E\left\{\sum_{\mathbf{m}}\sum_{\mathbf{q}} h(\mathbf{m})\, x(\mathbf{n}-\mathbf{m})\, h(\mathbf{q})\, x(\mathbf{n}-\mathbf{p}-\mathbf{q})\right\} + E\left\{d(\mathbf{n})\, d(\mathbf{n}-\mathbf{p})\right\}$$

$$= \sum_{\mathbf{m}}\sum_{\mathbf{q}} h(\mathbf{m})\, h(\mathbf{q}) E\left\{x(\mathbf{n}-\mathbf{m})\, x(\mathbf{n}-\mathbf{p}-\mathbf{q})\right\} + R_{dd}(\mathbf{p})$$

$$\therefore R_{yy}(\mathbf{p}) = \sum_{\mathbf{m}}\sum_{\mathbf{q}} h(\mathbf{m})\, h(\mathbf{q})\, R_{xx}(\mathbf{p}+\mathbf{q}-\mathbf{m}) + R_{dd}(\mathbf{p})$$

as

$$E\left\{x(\mathbf{n}-\mathbf{m})x(\mathbf{n}-\mathbf{p}-\mathbf{q})\right\} = E\left\{x(\mathbf{n}-\mathbf{m})x(\mathbf{n}-\mathbf{m}-(\mathbf{p}+\mathbf{q}-\mathbf{m}))\right\}$$

Now take the Fourier transform of each side to give:

$$P_{yy}(\boldsymbol{\omega}) = \sum_{\mathbf{p}} \left\{ \sum_{\mathbf{m}} \sum_{\mathbf{q}} h(\mathbf{m})\, h(\mathbf{q})\, R_{xx}(\mathbf{p}+\mathbf{q}-\mathbf{m}) \right\} e^{-j\omega^T \mathbf{p}} + P_{dd}(\boldsymbol{\omega})$$

where P_{dd} is the FT of the autocorrelation function of the noise. Interchange order;

$$P_{yy}(\boldsymbol{\omega}) = \sum_{\mathbf{m}} \sum_{\mathbf{q}} h(\mathbf{m})\, h(\mathbf{q}) \sum_{\mathbf{p}} R_{xx}(\mathbf{p}+\mathbf{q}-\mathbf{m})\, e^{-j\omega^T \mathbf{p}} + P_{dd}(\boldsymbol{\omega})$$

Let $\mathbf{k} = (\mathbf{p}+\mathbf{q}-\mathbf{m})$, then:

$$P_{yy}(\boldsymbol{\omega}) = \sum_{\mathbf{m}} \sum_{\mathbf{q}} h(\mathbf{m})\, h(\mathbf{q}) \sum_{\mathbf{k}} R_{xx}(\mathbf{k})\, e^{-j\omega^T (\mathbf{k}-\mathbf{q}+\mathbf{m})} + P_{dd}(\boldsymbol{\omega})$$

$$\therefore P_{yy}(\omega) = \left\{ \sum_{\mathbf{m}} h(\mathbf{m})\, e^{-j\omega^T \mathbf{m}} \right\} \left\{ \sum_{\mathbf{q}} h(\mathbf{q})\, e^{j\omega^T \mathbf{q}} \right\} \left\{ \sum_{\mathbf{k}} R_{xx}(\mathbf{k})\, e^{-j\omega^T \mathbf{k}} \right\} + P_{dd}(\omega)$$

$$\therefore P_{yy}(\boldsymbol{\omega}) = |H(\boldsymbol{\omega})|^2\, P_{xx}(\boldsymbol{\omega}) + P_{dd}(\boldsymbol{\omega}) \qquad (6)$$

as h is real.

The other term in the Wiener-Hopf equation is:

$$R_{xy}(\mathbf{p}) = E\{x(\mathbf{n})\, y(\mathbf{n}-\mathbf{p})\}$$

$$= E\left\{ \left[\sum_{\mathbf{m}} h(\mathbf{m})\, x(\mathbf{n}-\mathbf{p}-\mathbf{m}) + d(\mathbf{n}) \right] x(\mathbf{n}) \right\}$$

The image $x(\mathbf{n})$ and the noise $d(\mathbf{n})$ are uncorrelated and the noise has zero mean (as before):

$$\therefore R_{xy}(\mathbf{p}) = E\left\{ \sum_{\mathbf{m}} h(\mathbf{m})\, x(\mathbf{n}-\mathbf{p}-\mathbf{m})\, x(\mathbf{n}) \right\} = \sum_{\mathbf{m}} h(\mathbf{m}) E\left\{ x(\mathbf{n})\, x(\mathbf{n}-[\mathbf{p}+\mathbf{m}]) \right\}$$

$$= \sum_{\mathbf{m}} h(\mathbf{m})\, R_{xx}(\mathbf{p}+\mathbf{m})$$

Taking the Fourier transform of each side gives:

$$P_{xy}(\boldsymbol{\omega}) = \sum_{\mathbf{p}} \left\{ \sum_{\mathbf{m}} h(\mathbf{m})\, R_{xx}(\mathbf{p}+\mathbf{m}) \right\} e^{-j\boldsymbol{\omega}^T \mathbf{p}} = \sum_{\mathbf{m}} h(\mathbf{m}) \sum_{\mathbf{p}} R_{xx}(\mathbf{p}+\mathbf{m})\, e^{-j\boldsymbol{\omega}^T \mathbf{p}}$$

Let $\mathbf{p}+\mathbf{m}=\mathbf{k}$, then:

$$= \sum_{\mathbf{m}} h(\mathbf{m})\, e^{j\boldsymbol{\omega}^T \mathbf{m}} \sum_{\mathbf{k}} R_{xx}(\mathbf{k})\, e^{-j\boldsymbol{\omega}^T \mathbf{k}}$$

$$\therefore P_{xy}(\boldsymbol{\omega}) = H^*(\boldsymbol{\omega})\, P_{xx}(\boldsymbol{\omega})$$

Substituting back into the W-H equation gives:

$$\boxed{G(\boldsymbol{\omega}) = \frac{H^*(\boldsymbol{\omega})\, P_{xx}(\boldsymbol{\omega})}{|H(\boldsymbol{\omega})|^2\, P_{xx}(\boldsymbol{\omega}) + P_{dd}(\boldsymbol{\omega})}} \qquad (7)$$

This is the most commonly used form of the Wiener Filter.

NOTES on the Wiener Filter

1. At frequencies where:

$$|H(\boldsymbol{\omega})|^2\, P_{xx}(\boldsymbol{\omega}) \gg P_{dd}(\boldsymbol{\omega})$$

   the Wiener filter becomes the inverse filter $1/H(\boldsymbol{\omega})$.

2. If the noise is large there will be little deconvolution (ie. inverse filtering)

3. If the blurring function is unity (ie no blurring) then the Wiener filter simply recovers the image from the noise; the filter frequency response is then:

$$G(\boldsymbol{\omega}) = \frac{P_{xx}(\boldsymbol{\omega})}{P_{xx}(\boldsymbol{\omega}) + P_{dd}(\boldsymbol{\omega})}$$

NOTES continued

We should also address the question of how one might actually go about estimating P_{xx} (the power spectrum of the original image) and P_{dd} (the power spectrum of the noise).

1. to estimate P_{dd} we might choose a section of the image which is effectively featureless and form its power spectrum – we are then forming the PS of the noise. Generally this works well in many cases (note that if we have white noise the power spectrum, P_{dd}, is constant).

2. estimating P_{xx} is clearly harder as we are actually trying to recover the real image – so what might we do in practice? One common method is to approximate P_{xx} with a constant, K. K can be chosen interactively to produce the empirically best results.

The FIR Wiener Filter

Note, in general, we get a filter with an infinite region of support. Can derive an FIR approximation but this would not be the optimum FIR filter.

Repeating derivation of W-H equations shows that they are also valid for a finite support region R_G:

$$R_{xy}(\mathbf{p}) = \sum_{\mathbf{q}} g(\mathbf{q})\, R_{yy}(\mathbf{p}-\mathbf{q}) \quad \forall\, \mathbf{p},\mathbf{q} \in R_G$$

Specifically if the region of support is:
$R_G = \{p_1, p_2;\ -M \le p_1 \le M,\ -M \le p_2 \le M\}$ the filter has $(2M+1)^2$ taps or impulse response coefficients and W-H equations are

$$\boxed{\begin{array}{c} R_{xy}(p_1, p_2) = \displaystyle\sum_{q_2=-M}^{M} \sum_{q_1=-M}^{M} g(q_1, q_2) R_{yy}(p_1 - q_1, p_2 - q_2) \\ \forall\, \mathbf{p} \in -M \leq p_1 \leq M,\ -M \leq p_2 \leq M\} \end{array}}$$

Can express in matrix form and solve for the filter weights $g(q_1, q_2)$ if we can estimate R_{xy} (and perhaps measure R_{yy}).

Results

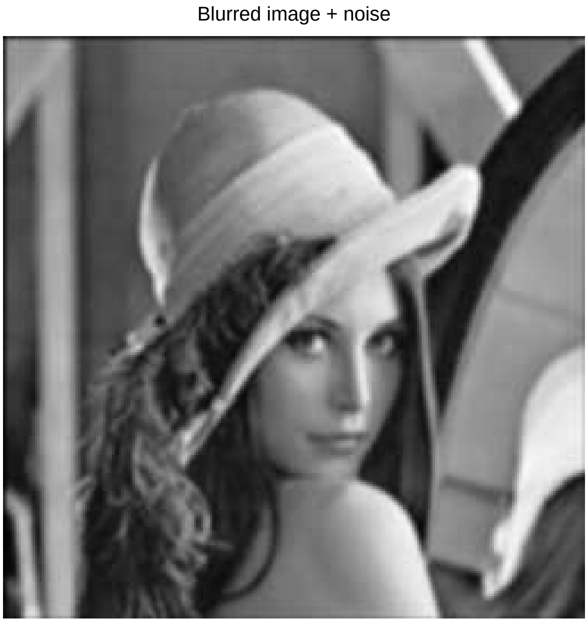

Blurred image + noise

Figure 7: Blurred image with small additive noise

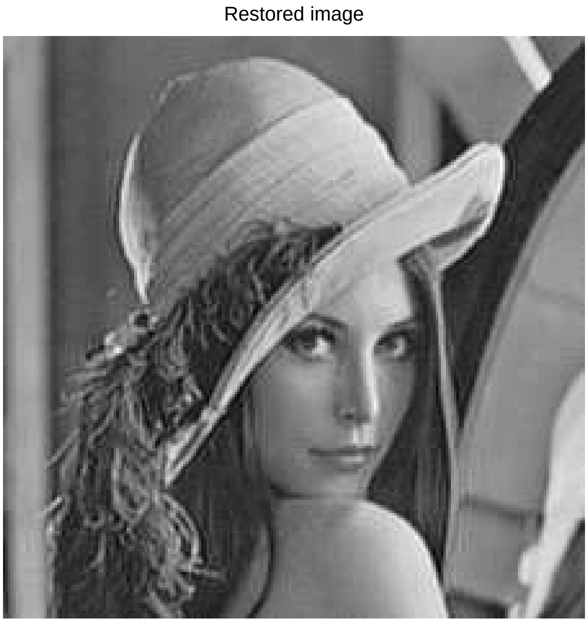

Restored image

Figure 8: Image restored by Wiener filter

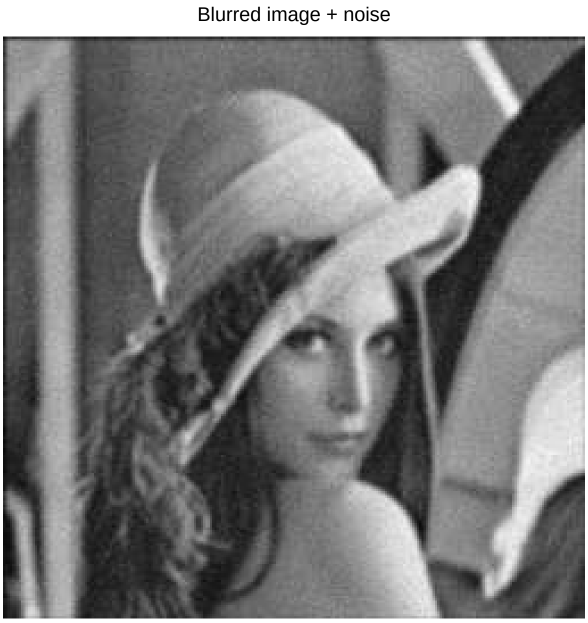


Figure 9: Blurred image with increased additive noise

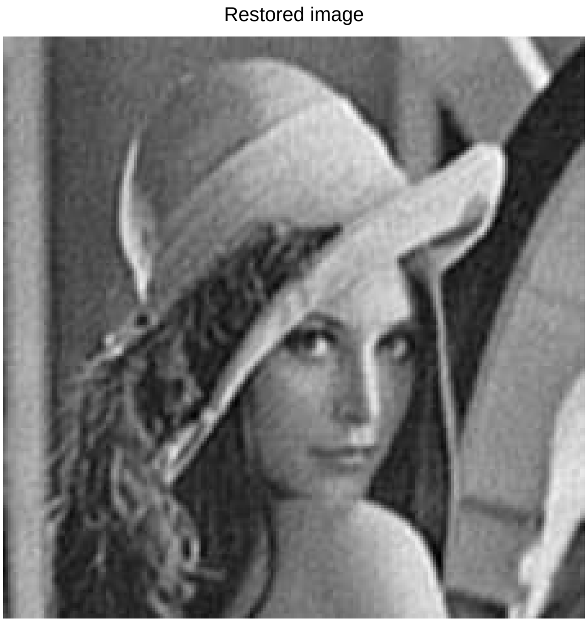


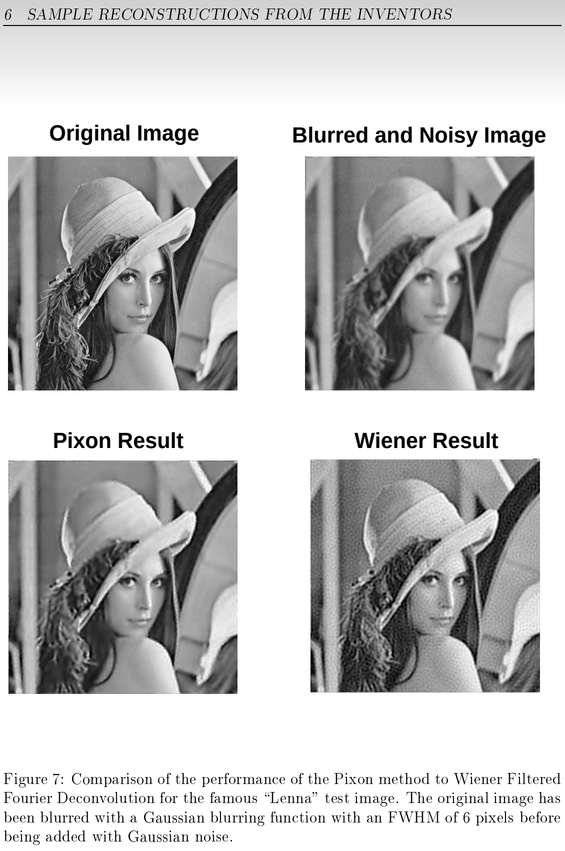
Figure 10: Image restored by Wiener filter

It is apparent that the Wiener filter only works well when there is a small amount of noise – if the image is significantly noisy the method does not provide sufficient regularisation resulting in artefacts in the restored image.

Note that the above images have been created using the Wiener filter where P_{xx} and P_{dd} were known because we had knowledge of both our original image and our noise – in cases where we do not have this knowledge, we must form the Wiener filter using *estimated* values of P_{xx} and P_{dd}, and the results are generally poor.

Despite the often poor results of the Wiener filter when used on images which have significant blurring and noise, it is one of the basic image processing techniques which has provided a starting point for many more sophisticated methods.

We showed that the Wiener filter was the *optimal filter*, why then is its performance so poor? The answer is that it is optimal over the space of *linear* filters, we will show that by extending our space to *non-linear* filters, we can often do much better.



6 SAMPLE RECONSTRUCTIONS FROM THE INVENTORS

Figure 7: Comparison of the performance of the Pixon method to Wiener Filtered Fourier Deconvolution for the famous "Lenna" test image. The original image has been blurred with a Gaussian blurring function with an FWHM of 6 pixels before being added with Gaussian noise.

Figure 11: Page taken from *The Pixon Method of Image Reconstruction R. C. Puetter & A. Yahil, in Astronomical Data Analysis Software and Systems VIII, D. M. Mehringer, R. L. Plante & D. A. Roberts, eds., ASP Conference Series, 172, pp. 307-316, (1999).*

IIB 4F8: Image Processing and Image Coding

Handout 6 : Image Deconvolution cont. and Image Enhancement

J Lasenby

Signal Processing Group,
Engineering Department,
Cambridge, UK

Lent 2016

Spectral Estimation

We note that the Wiener filter relies on the fact that we know or can accurately estimate the power spectra of the original signal and the noise, P_{xx} and P_{dd}.

A reasonable estimate of the power spectrum of the noise can often be achieved by isolating a portion of the image that we think contains only noise (for example where the original signal is uniform, if such a region exists)

– we can then take the FT of that region (suitably windowed) and form the power spectrum from $|FT|^2$.

Estimating the power spectrum of the signal is not such a simple matter; one method would be as follows: assume we have estimated P_{dd}.

$$y(\mathbf{n}) = Lx(\mathbf{n}) + d(\mathbf{n})$$

then taking FTs gives

$$Y(\omega) = H(\omega)X(\omega) + D(\omega)$$

where $H = FT(L)$. Thus, if we have an estimate of D, we can estimate $Y - D$ to be approximately the FT of the signal. Or, if our blurring function is known we can obtain a better estimate of X by pseudo-inverse filtering, i.e.

$$X(\omega) = \frac{Y(\omega) - D(\omega)}{H_g(\omega)}$$

where H_g is the pseudo-inverse filter formed from H (handout 5).

What we would really like is an iterative method whereby we could make initial guesses/estimates of our signal and noise power spectra, do the deconvolution and then from the results (looking at residuals etc.) modify our power spectra estimates.

However, since the Wiener filter is a linear method, it does not lend itself to such an iterative minimisation

– to do a better job we need to go to non-linear filters. By way of introduction to one of the main types of non-linear methods, we now look at deriving the Wiener filter from a Bayesian perspective.

Bayesian derivation of Wiener Filter

Now sketch an alternative derivation of the Wiener filter – can then extend the method to indicate how we might construct non-linear filters which will do a much better job than the Wiener filter.

Suppose, as before, that our observed image, y, the original image, x, the linear distortion L, and the noise d, are related by

$$y(\mathbf{n}) = Lx(\mathbf{n}) + d(\mathbf{n})$$

Write this equation in vector form –

$$\mathbf{y} = L\mathbf{x} + \mathbf{d}$$

For simplicity in the derivations we shall assume that $E[\mathbf{x}] = 0$ and $E[\mathbf{d}] = 0$, i.e. that both the signal and the noise are zero mean.

To find an estimate of $\mathbf{x}$, we choose to maximise $Pr(\mathbf{x}|\mathbf{y})$, i.e. the *probability* of the original image *given* the observed data. When dealing with *conditional probabilities* we use *Bayes' Theorem*:

$$Pr(\mathbf{x}|\mathbf{y}) = \frac{1}{Pr(\mathbf{y})} Pr(\mathbf{y}|\mathbf{x}) Pr(\mathbf{x})$$

sometimes also written as $Pr(\mathbf{x}|\mathbf{y})Pr(\mathbf{y}) = Pr(\mathbf{y}|\mathbf{x})Pr(\mathbf{x})$. In words we write

$$\text{Posterior probability} = \frac{1}{\text{evidence}}[\text{likelihood}][\text{prior probability}]$$

Can regard $Pr(\mathbf{y})$, the probability of the data, simply as a normalising factor, which therefore implies that we wish to maximise

$$Pr(\mathbf{x}|\mathbf{y}) \propto Pr(\mathbf{y}|\mathbf{x}) Pr(\mathbf{x})$$

Assume gaussian noise (although there is no problem in including a knowledge of the noise – for example, Poisson noise is common in optical problems – we will here derive our equations for gaussian noise). Can write the probability of the noise, which is proportional to the likelihood as

$$Pr(\mathbf{y}|\mathbf{x}) \propto e^{-\frac{1}{2}\mathbf{d}^T N^{-1}\mathbf{d}} = e^{-\frac{1}{2}(\mathbf{y}-L\mathbf{x})^T N^{-1}(\mathbf{y}-L\mathbf{x})}$$

where $N = E[\mathbf{dd}^T]$ is the noise covariance matrix. The $\mathbf{d}^T N^{-1}\mathbf{d}$ term is the vector equivalent of the $\frac{1}{\sigma^2}$ term in the 1d gaussian – if N is diagonal then N^{-1} will be diagonal with elements $\frac{1}{\sigma_i^2}$.
Usually write

$$Pr(\mathbf{y}|\mathbf{x}) \propto e^{-\frac{1}{2}\chi^2}$$

where $\chi^2 = (\mathbf{y} - L\mathbf{x})^T N^{-1}(\mathbf{y} - L\mathbf{x})$.

Now have to decide on the assignment of the *prior* probability $Pr(\mathbf{x})$ – this probability incorporates any prior knowledge we may have about the distribution of the data.
Assume an ideal world in which $\mathbf{x}$ is a gaussian random variable, described by a *known* covariance matrix $C = E[\mathbf{xx}^T]$ so that

$$Pr(\mathbf{x}) \propto e^{-\frac{1}{2}\mathbf{x}^T C^{-1}\mathbf{x}}$$

Thus we can now write the posterior probability as

$$Pr(\mathbf{x}|\mathbf{y}) \propto Pr(\mathbf{y}|\mathbf{x})Pr(\mathbf{x}) \propto e^{-\frac{1}{2}(\chi^2+\mathbf{x}^T C^{-1}\mathbf{x})}$$

which one must maximise wrt $\mathbf{x}$ to obtain the reconstruction.
This is equivalent to minimising $F = \frac{1}{2}(\chi^2 + \mathbf{x}^T C^{-1}\mathbf{x})$.

In fact, completing the square in $\mathbf{x}$ enables us to write the posterior as

$$Pr(\mathbf{x}|\mathbf{y}) \propto e^{-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T M^{-1}(\mathbf{x}-\hat{\mathbf{x}})} \qquad (1)$$

where

$$\begin{aligned} \hat{\mathbf{x}} &= W\mathbf{y} = \text{the image reconstruction} \\ W &= (C^{-1} + L^T N^{-1} L)^{-1} L^T N^{-1} \end{aligned} \qquad (2)$$

this is in fact the *Wiener filter* matrix! and

$$M = (C^{-1} + L^T N^{-1} L)^{-1} \qquad (3)$$

is the reconstruction error matrix $E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T]$.

Below we verify that the above results are correct by first substituting for W into equation 1

$$(\mathbf{x} - W\mathbf{y})^T M^{-1}(\mathbf{x} - W\mathbf{y}) = \mathbf{x}^T M^{-1}\mathbf{x} - \mathbf{x}^T M^{-1} W\mathbf{y} - \mathbf{y}^T W^T M^{-1}\mathbf{x} + \mathbf{y}^T W^T M^{-1} W\mathbf{y} \qquad (4)$$

Now show that the terms containing $\mathbf{x}$ in the above can be equated to the terms containing $\mathbf{x}$ in F

$$F = (\mathbf{y}^T - \mathbf{x}^T L^T) N^{-1} (\mathbf{y} - L\mathbf{x}) + \mathbf{x}^T C^{-1}\mathbf{x}$$

This implies that

$$\mathbf{x}^T M^{-1}\mathbf{x} = \mathbf{x}^T L^T N^{-1} L\mathbf{x} + \mathbf{x}^T C^{-1}\mathbf{x}$$

Verifying that

$$M = (C^{-1} + L^T N^{-1} L)^{-1} \qquad (5)$$

Next, equating the terms in $\mathbf{x}^T(..)\mathbf{y}$ gives us

$$M^{-1}W = L^T N^{-1}$$

and similarly equating the terms in $\mathbf{y}^T(..)\mathbf{x}$ gives

$$N^{-1}L = W^T M^{-1}$$

These expressions are consistent with equations 2,3 provided $(N^{-1})^T = N^{-1}$ and $(M^{-1})^T = M^{-1}$. It is easy to see that $(N^{-1})^T = N^{-1}$ is indeed true from the fact that $N = E(\mathbf{dd}^T)$, and as it is also clear that $(C^{-1})^T = C^{-1}$ since $C = E(\mathbf{xx}^T)$, equation 5 tells us that $(M^{-1})^T = M^{-1}$.

If we write our previous form of the Wiener filter as

$$G(\omega) = \frac{H^*(\omega)(P_{dd}(\omega))^{-1}}{|H(\omega)|^2(P_{dd}(\omega))^{-1} + (P_{xx}(\omega))^{-1}}$$

and compare this to

$$W = (C^{-1} + L^T N^{-1} L)^{-1} L^T N^{-1}$$

we are able to see the equivalence of $G(\omega)$ and W.

Thus we have recovered the optimal *linear* method via a Bayesian analysis whereas previously we had obtained it by minimising residual variances.

Now note the remarkable feature of this: original problem was

$$\mathbf{y} = L\mathbf{x} + \mathbf{d}$$

In general L will *not be invertible*. e.g. may be a blurring (beam) function in which higher spatial frequencies are strongly suppressed.

However, the estimate $\hat{\mathbf{x}} = W\mathbf{y}$ can still be computed no matter how singular L is — only needs L^T to be evaluated!

This is an example of **Regularization**.

Notice how if the C^{-1} were not present in W we would just have $W = L^{-1}$ [since with no C^{-1} we can write $W = (L^T N^{-1} L)^{-1} (L^T N^{-1} L) L^{-1} = L^{-1}$]

— we say we have *regularized* the inverse; effectively we have added on another term in order to avoid singlularities.

Maximum Entropy Method (MEM) deconvolution

The Wiener solution is 'easy' to calculate and has known reconstruction errors. However, it is certainly by no means the best in real problems. It depends on the *assumption* of gaussianity and knowledge of the covariance structure *a priori*.

Since the world this not so simple, we are forced to consider alternative *priors*. One such prior which has been widely and successfully used is the *entropy prior*.

Usually MEM is applied to positive, additive distibutions (PADS). Let $\mathbf{x}$ be the (true) pixel vector we are trying to estimate. A consideration of the following general conditions;

- subset independence
- coordinate invariance
- system independence

leads to the following prior.....

$$Pr(\mathbf{x}) \propto e^{\alpha S}$$

where the *cross entropy* S of the image is given by

$$S(\mathbf{x}, \mathbf{m}) = \sum_i \left[x_i - m_i - x_i \ln\left(\frac{x_i}{m_i}\right)\right]$$

where $\mathbf{m}$ is the *measure* on an image space (*the model*) to which the image $\mathbf{x}$ defaults in the absence of data. (Can see global maximum of S occurs at $\mathbf{x} = \mathbf{m}$.)

Must then maximise the posterior distribution with the above entropic prior for $Pr(\mathbf{x})$.

As this is a complicated non-linear equation, the optimization must be performed via some sort of iterative technique. The results are often impressive and the MEM method has been used extensively in astronomy, medical imaging and general image processing.

The Pixon Method

The Pixon method is another recently established deconvolution technique – derived initially to deconvolve astronomical images.

We will not look at this in any detail but will state that it can also be derived in a Bayesian manner as outlined above using a different prior. The aim of the pixon method is to first find a distribution of *pixons* across the image – a pixon is a box of varying size, the smallest being a single pixel. We will have large pixons where the image has little detail and small pixons where there is fine detail.

These pixons are then our basis elements of our image and we set up and minimise our posterior distribution in terms of them. The results of the Pixon method have been very impressive (but note that when the figures below compare with MEM it is generally a poor version of MEM).

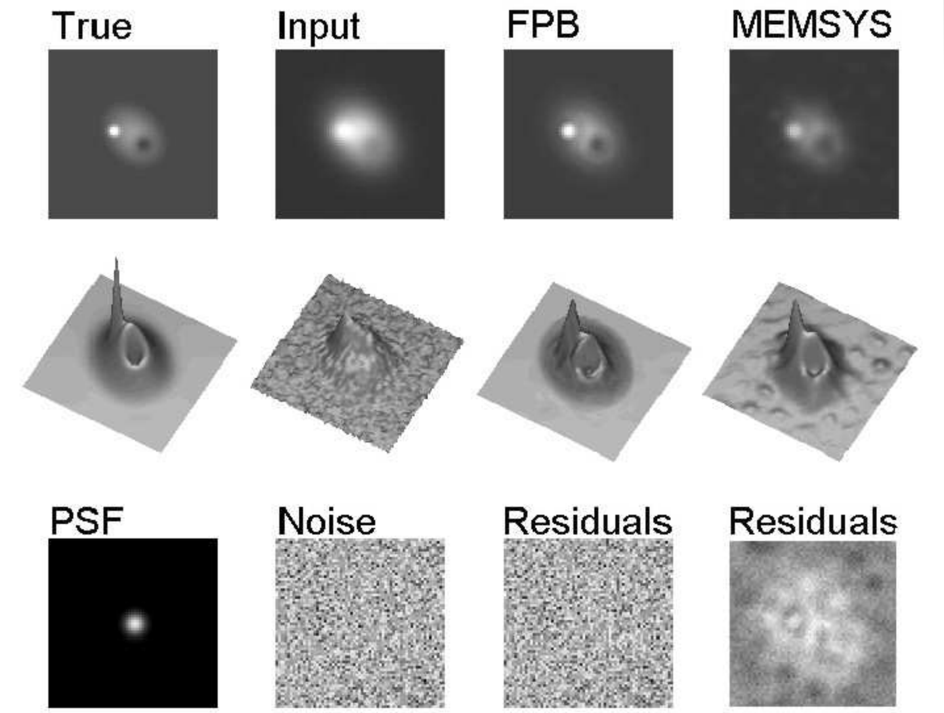
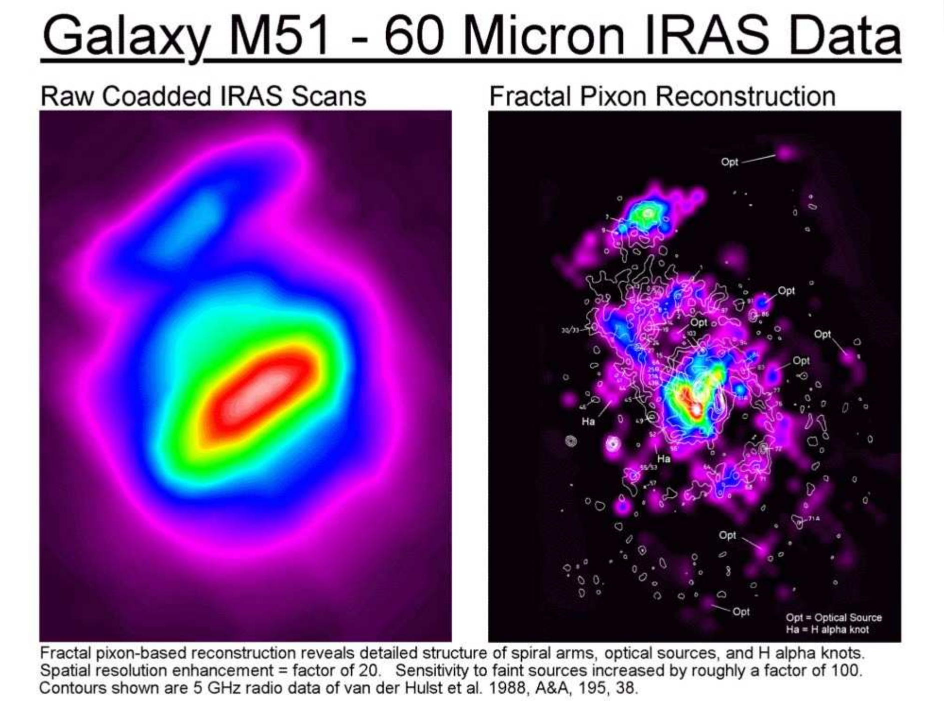


Figure 1: A comparison of three deconvolution techniques on simulated data

Figure taken from http://casswww.ucsd.edu/personal/puetter/pixonpage.html.



Fractal pixon-based reconstruction reveals detailed structure of spiral arms, optical sources, and H alpha knots. Spatial resolution enhancement = factor of 20. Sensitivity to faint sources increased by roughly a factor of 100. Contours shown are 5 GHz radio data of van der Hulst et al. 1988, A&A, 195, 38.

Figure 2: The Pixon method deconvolution of galaxy data

Figure taken from http://casswww.ucsd.edu/personal/puetter/pixonpage.html.

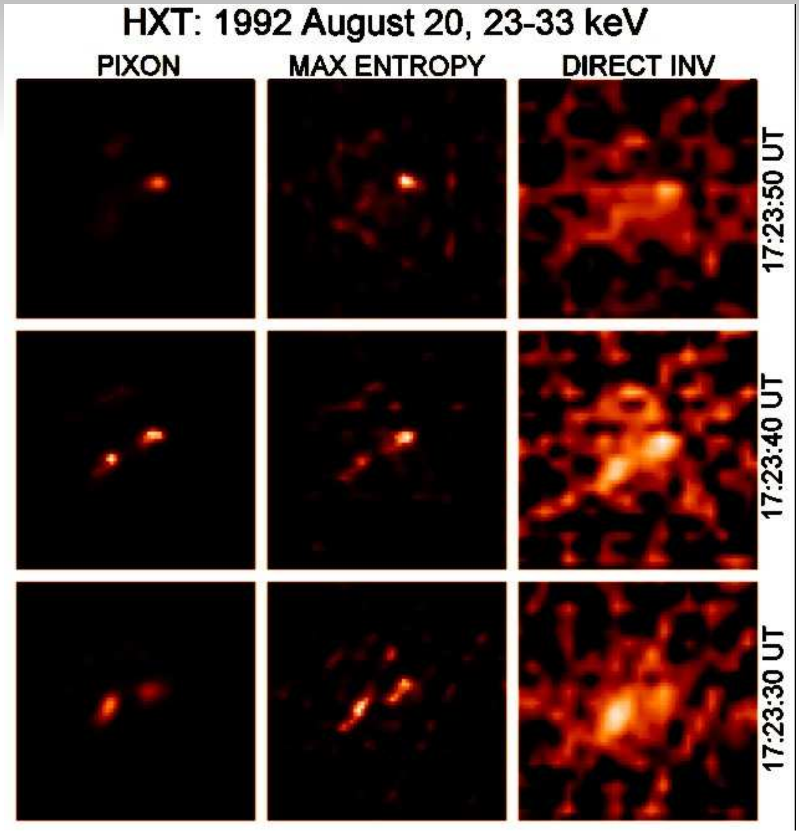


Figure 3: The Pixon method deconvolution of x-ray data

Figure taken from
http://casswww.ucsd.edu/personal/puetter/pixonpage.html.

Figures 1,2,3 show various applications of the Pixon method on real and simulated images. Figure 1 is particularly interesting as it shows that the pixon reconstruction not only produces a good likeness of the original, but also produces residuals which appear to have the same statistics as the added noise.

Looking at the statistics of the residuals is a very good way of determining how well the deconvolution technique is performing (although in practice we may not of course know the statistics of the noise). We would like to see as little structure in the residuals as possible.

The hardest deconvolution is *blind deconvolution* when we do not know the point spread function.

This can be attacked in a number of ways. In MEM and Pixon techniques we can assume a form for the PSF but give it a variable width (e.g. gaussian with a variance which becomes a parameter we search over).

Another 'recent' development in blind deconvolution is the *ensemble learning* method – this too adopts a Bayesian approach and can produce impressive results (if interested see *Miskin and Mackay, Ensemble learning for blind image separation and deconvolution. Advances in independent component analysis, (Ed Girolami), Springer Verlag, July 2000*).

Compressed Sensing – in a compressed form

A currently very fashionable area of Signal Processing is known as *compressed sensing*.

Look briefly at what this is:

- Suppose we have underlying real data $\bar{\mathbf{x}}$ and that $\bar{\mathbf{x}}$ is sparse in some basis
- Let $\mathbf{x}$ be the sparse representation in some basis Ψ, $\mathbf{x} = \Psi\bar{\mathbf{x}}$.
- Take another *random* or *incoherent* basis Φ, and form $\mathbf{y} = \Phi\mathbf{x}$.
- Task is to reconstruct the signal $\mathbf{x}$ from these random/incoherent projections

Compressed Sensing, continued

- We look for the sparsest solution – the $\mathbf{x}$ for the smallest l_0 norm. But this is hard.
- Some clever maths then tells us we can recover the signal via minimising the l_1 norm
- ...and if we do this, and if some conditions are satisfied, we recover the original signal with high probability.
- So, this seems as though we can redefine our sampling criteria if we know something about our signal.
- But very little to date which is impressive in the deconvolution of images field using compressed sensing.

Contrast

Consider the illustrations in the figure.

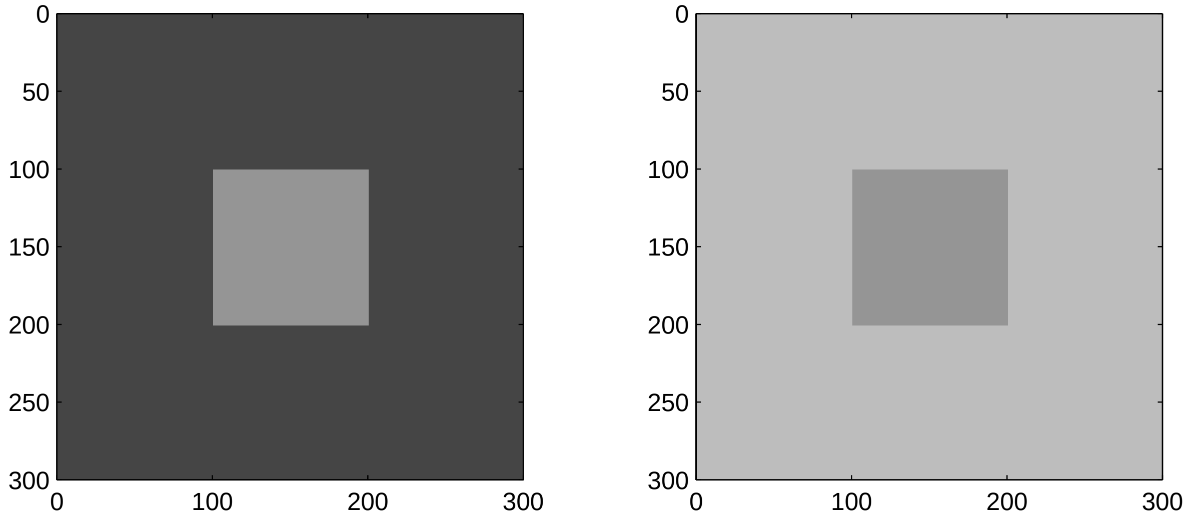


Figure 4: Illustration of *contrast*: inner squares have equal luminances/intensities but do not appear equally bright

Now consider the figure below:

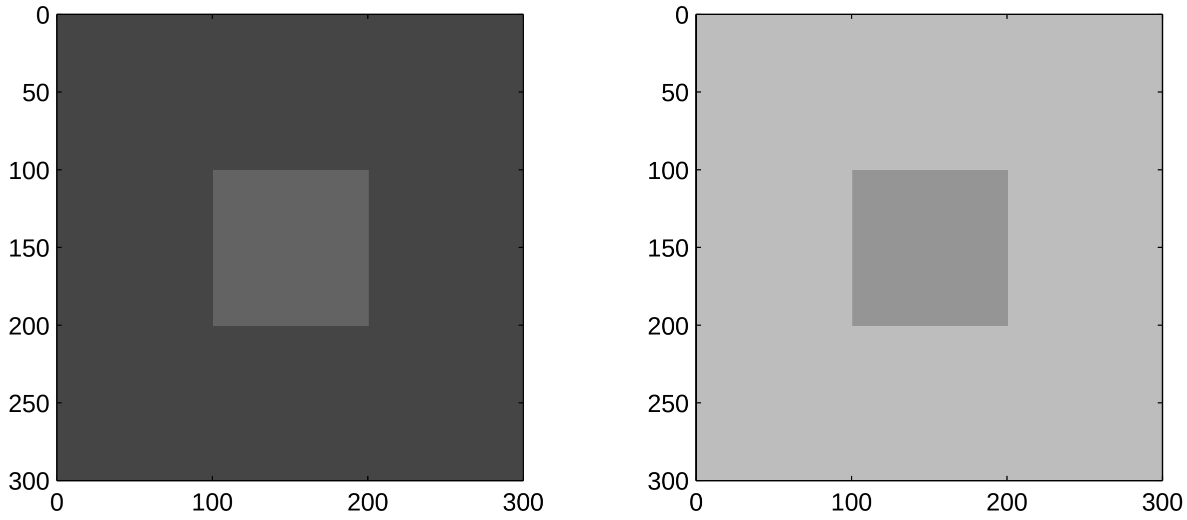


Figure 5: Illustration of *contrast*: inner squares have unequal luminances/intensities but appear equally bright

Weber's Law states that for the intensity of an object, f_0 to be such that it changes from being indiscernible from the background to just noticeably different to the background, which has intensity f_b, the ratio of the difference to the object's intensity must be of a particular constant value

$$\frac{|f_b - f_0|}{f_o} = \text{constant}$$

If $f_0 = f$ and $f_b = f + \Delta f$ where Δf is the small change to produce the noticeable difference in intensity, then

$$\frac{\Delta f}{f} \simeq d(\log f) = \Delta c$$

since $\frac{d(\log f)}{df} = \frac{1}{f}$. Δc is a constant (empirically taking value of 0.02).

Previous equation $\implies$ equal increments in the log of the intensity should be perceived as equal increments in contrast ($\Delta(\log f) \propto \Delta c$).

Thus $\frac{dc}{d \log f} = k_1$, k_1 constant, so that

$$c = k_1 \log f + k_2$$

This is the *logarithmic law* of intensity to contrast. The other widely used law is a power law, where the relationship between the contrast and the intensity can be written as

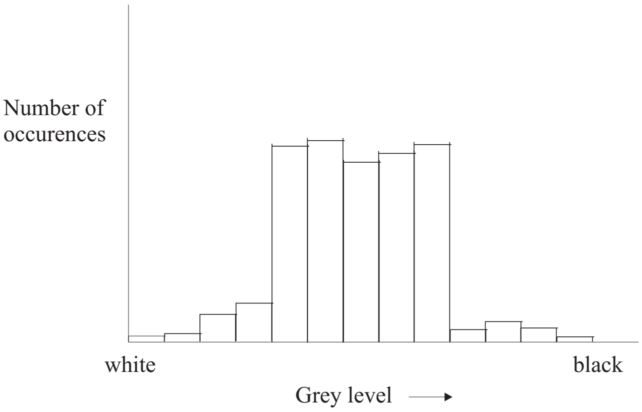
$$c = \alpha_n f^{\frac{1}{n}} \quad \text{or} \quad f \propto c^n$$

where n is often taken to be 3.

Histogram Equalisation

In some cases images may exhibit poor use of the available grey levels; may result from poor illumination of the scene or non-linearity in the imaging system.

To examine this, plot a histogram of frequency of occurrence of grey level against grey level as shown below.



Here most of the image pixels are concentrated around the mid-range grey levels and little use is made of the extreme black and white levels.

An intuitively appealing idea would be to apply a transformation or mapping to the image pixels such that the probability of occurrence of the various grey levels should be constant

i.e. all grey levels are equiprobable, which would correspond to a constant amplitude histogram.

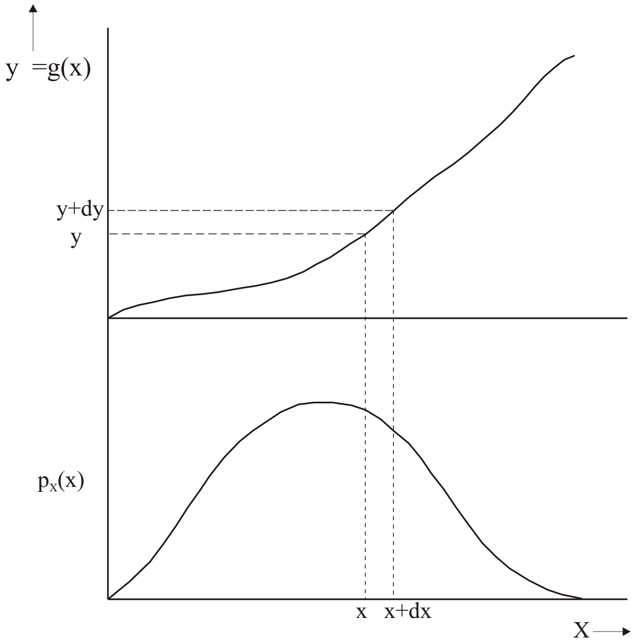
Assume that the image pixel levels are mapped by the function

$$y = g(x)$$

where x is an image pixel level and y is the transformed level.

Let the probability density of the pixels in the image be $p_X(x)$ and the probability density of the output image be $p_Y(y)$. (X, Y, random variables).

Note that the probability of X lying between x and $x + \delta x$, must be the same as the probability of $g(X) = Y$ lying between y and $y + \delta y$,



$$Pr\{x \leq X < x + dx\} = Pr\{y \leq Y < y + dy\}$$

$$p_Y(y)dy = p_X(x)dx \quad \Longrightarrow \quad p_Y(y) = \frac{p_X(x)}{\frac{dy}{dx}}$$

It is required that the output image probability density $p_Y(y)$ be constant over the grey level range O to L.

$$\therefore \; p_Y(y) = \frac{1}{L}$$

$$\frac{dy}{dx} = Lp_X(x) \quad \Longrightarrow \quad y = g(x) = \int_0^x L\, p_X(u)du$$

In practice the input image probability density is not known and is approximated by the image histogram and the integral is approximated by a sum.

Let the input image by quantised into N_L levels x_i spaced by Δx_i then $N_L \Delta x_i = L$ and

$$y_k = \sum_{i=1}^{k} Lp_X(x_i)\Delta x_i \;\; \text{for} \;\; k = 1, .., N_L$$

Now

$$p_X(x_i)\Delta x_i = Pr\{x_i \leq X \leq x_i + \Delta x_i\}$$

so if the histogram of the image has N_i occurrences in the bin x_i to $x_i + \Delta x_i$ then

$$p_X(x_i)\Delta x_i = Pr\{x_i \leq X \leq x_i + \Delta x_i\} = \frac{N_i}{N \times M}$$

where N and M are the dimensions of the image in pixels. Thus the mapping rule becomes:

$$y_k = \sum_{i=1}^{k} L\frac{N_i}{NM}$$

Note that if $k = N_L$, $y_{N_L} = L$ as required. This may be considered as a look-up table –i.e. the above values of y_k are formed and stored so that we can scan our image x and when the value of pixel i in x falls within the kth greylevel bin, we map it to y_k.

Practical Notes

1. Matlab has a function `hist` which will perform the histogram operation and a function `cumsum` which forms the cumulative sum required in evaluating the mapping.

2. If relatively few bins are used for the histogram then the look-up table may require interpolation to give a smoother function.

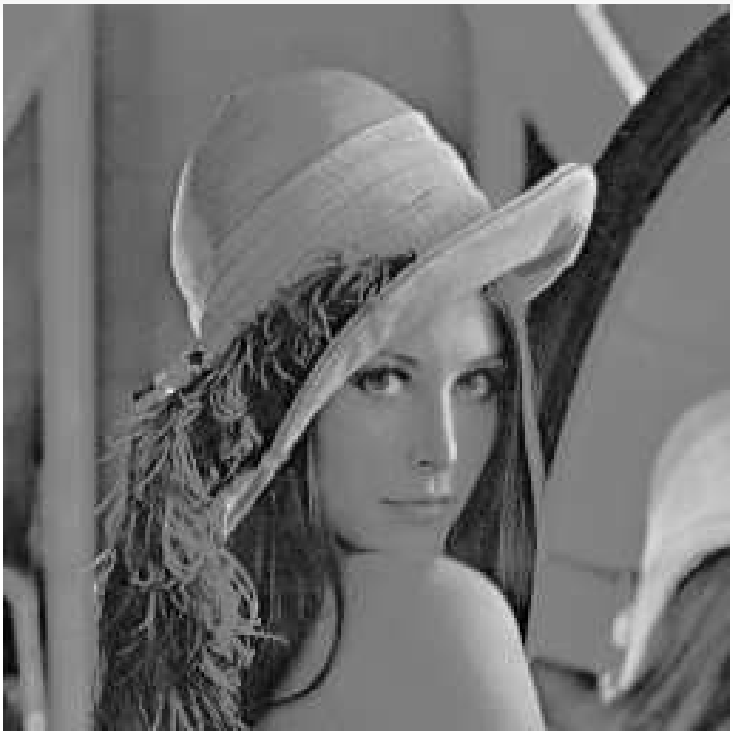


Figure 7: Image with poor distribution of brightness levels

In figure 7 we have the Lenna image with a poor distribution of brightness levels. This image has been formed by *contrast mapping* according to the following formula:

$$y = f(x) = k + \mathrm{sign}(x-k)|x-k|\left[\frac{|x-k|}{a}\right]^p$$

where $k = \frac{1}{2}(x_{min} + x_{max})$ is the average value of the image intensity, $a = \frac{1}{2}(x_{max} - x_{min})$ and sign is $+1, 0, -1$ if $(x-k)$ is positive, zero or negative.

The exponent p was taken as 2. Note that this mapping preserves both the maximum, minimum and average values of the image.

Figure below shows the previous figure after histogram equalisation using 64 greylevels. We can see that the greylevels have been 'evened out' so that the image is now more aesthetically pleasing.

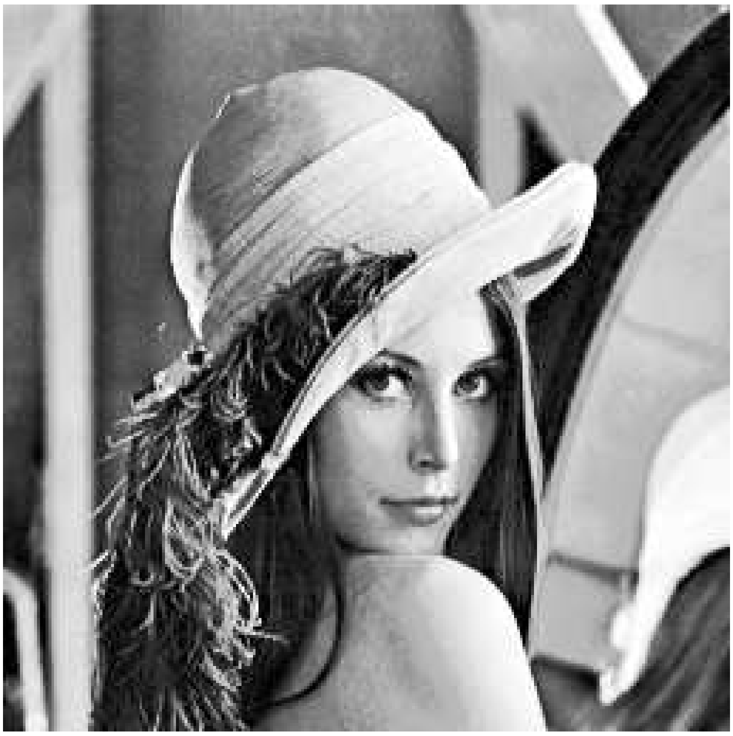


Figure 8: Contrast mapped Lenna image with Histogram Equalisation

Example: Taken from Part IIB Tripos 2006

2. (a) Explain, qualitatively, the concept of *histogram equalisation* in images. In what situations would histogram equalisation be useful? [10%]

Assuming that we have a range of greyscale values from 1 to 8, consider the 4×4 image given in figure 6

1	1	2	2
3	2	4	3
2	4	3	2
2	3	1	1

Sketch and comment on the histogram of this image.
Perform histogram equalisation on this image by finding the set of transformed values $\{y_k\}$, $k = 1, .., 8$ onto which the original greylevels are mapped.
Sketch the new equalised image and its histogram, commenting on how well the process has worked. [40%]

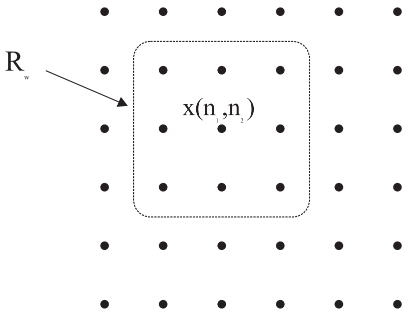
Median Filtering

Median filtering is essentially a method for detecting and replacing pixels which are "unlikely" in some sense.

Each pixel in the image is replaced by the median value of the pixels in a window containing the pixel under consideration. If the input image is $x(n_1, n_2)$ then the median filtered image $y(n_1, n_2)$ is given by:

$$y(n_1, n_2) = \underset{m_1, m_2 \in R_w}{\text{median}} \{x(n_1 - m_1, n_2 - m_2)\}$$

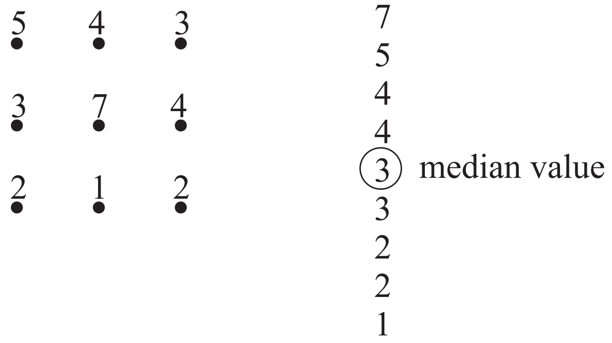
where R_w is the support region for the median filter.



The window is scanned over the whole image and the pixel $x(n_1, n_2)$ at the centre of the window is replaced by the median value of the pixels contained in the window surrounding pixel $x(n_1, n_2)$.

To form the median, order window pixels in ascending (or descending) brightness and choose centre value.

e.g. if the pixel values were as shown below, then the pixel at the centre of the window, value 7, would be replaced with a level of 3 which is the median.



5 4 3

3 7 4

2 1 2

7
5
4
4
3 median value
3
2
2
1

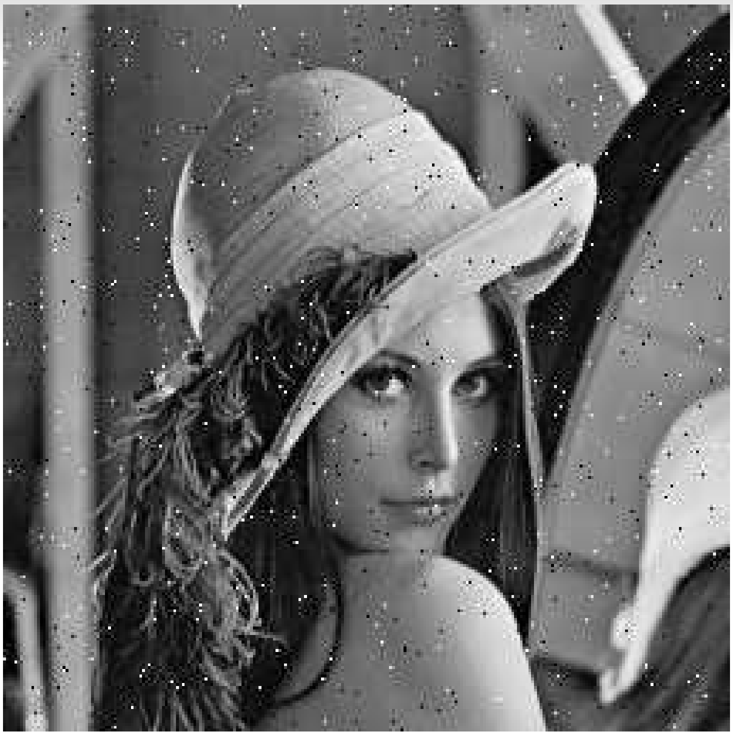


Figure 9: Image with binary valued impulsive noise

Note that this type of degradation can be performed by the Matlab command `J = imnoise(I,‘salt & pepper’, D)`, where binary valued noise is added to image I at a noise density of D to form new image J.

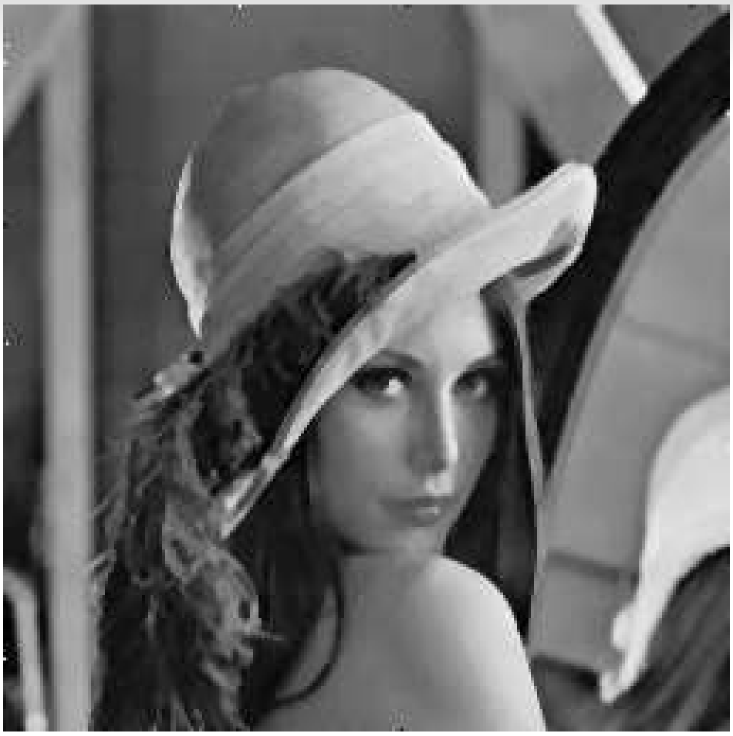


Figure 10: Image after median filtering with 3×3 window

Figure 10 shows the previous image after application of a median filter with 3×3 window. Note that the impulsive noise has disappeared but that there is a loss of detail due to the smoothing effects of the filter.

The 2d z-transform

We have discussed the 2d Fourier transform in some detail. However, we should note that the FT converges uniformly only for stable sequences. As an illustration, let us look at the 1d unit step: $h(t) = 1 \text{ if } t \geq 0, \quad h(t) = 0 \text{ if } t < 0$.

$$H(\omega) = \int_{-\infty}^{+\infty} h(t) e^{-j\omega t} dt = \int_{0}^{\infty} e^{-j\omega t} dt$$

which does not converge. Thus the spectrum of the step function cannot rigorously be expressed as a FT.

The z-transform converges for a much wider class of signals – it is also very useful (as you will have seen in the 3rd year courses) in studying difference equations and stability.

The 2d z-transform of an image or 2d function $x(u_1, u_2)$ is denoted by $X(z_1, z_2)$ and defined as a straightforward extension of the 1d z-transform as

$$X(z_1, z_2) = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} x(n_1, n_2) {z_1}^{-n_1} {z_2}^{-n_2}$$

where z_1 and z_2 are complex variables. The space of (z_1, z_2) is therefore 4-dimensional. It is easy to see that if we restrict z_i to lie on the unit circle, i.e. $z_i = e^{j\omega_i}$, then the z-transform reduces to the Fourier transform:

$$X(z_1, z_2)|_{z_1=e^{j\omega_1}, z_2=e^{j\omega_2}} = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} x(n_1, n_2) e^{-jn_1\omega_1} e^{-jn_2\omega_2} = FT(\omega_1, \omega_2)$$

the z-transform is therefore a generalisation of the Fourier transform. The 2d surface in the 4d space on which the FT lies ($z_i = e^{j\omega_i}$) is called the unit surface.

An important concept when looking at z-transforms is that of the region of convergence or RoC.
Consider evaluating $X(z_1, z_2)$ along $z_i = r_i e^{j\omega_i}$, where r_i, ω_i are radius and argument in the z_i planes. So, we therefore have

$$X(z_1, z_2)|_{z_i=r_i e^{j\omega_i}} = \sum_{n_1} \sum_{n_2} x(n_1, n_2) r_1^{-n_1} r_2^{-n_2} e^{-jn_1\omega_i} e^{-jn_2\omega_i}$$

which we recognise as the FT of $x(n_1, n_2) r_1^{-n_1} r_2^{-n_2}$.
Recall that uniform convergence of our FT, $FT(\omega_1, \omega_2)$, requires the absolute summability of $x(n_1, n_2)$, i.e.

$$\sum_{n_1} \sum_{n_2} |x(n_1, n_2)| < \infty$$

An absolutely summable sequence is a stable sequence: stable implies; bounded input $\Longrightarrow$ bounded output.
Similarly, uniform convergence of the z-transform requires the absolute summability of $x(n_1, n_2)r_1^{-n_1}r_2^{-n_2}$, i.e.

$$\sum_{n_1}\sum_{n_2}|x(n_1, n_2)|r_1^{-n_1}r_2^{-n_2} < \infty$$

absolute summability will therefore depend upon the values of r_1 and r_2.

One of the motivations for studying the z-transform is that it is a useful tool for the investigation of discrete-time, sampled data systems. In particular, it easily handles up and down sampling. To illustrate this look at the 1d z-transform of a signal $x(n)$;

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n)z^{-n}$$

Since $X(z^{\frac{1}{2}}) = \sum_{n=-\infty}^{+\infty} x(n)z^{-\frac{n}{2}}$ and
$X(-z^{\frac{1}{2}}) = \sum_{n=-\infty}^{+\infty} x(n)(-z)^{-\frac{n}{2}}$, then we can see that

$$\frac{1}{2}\left(X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})\right)$$

contains only the even terms, so that

$$\frac{1}{2}\left(X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})\right) = \sum_{n=-\infty}^{+\infty} x(2n)z^{-n}$$

which is the z-transform of the downsampled signal.

Similarly,

$$X(z^2) = \sum_n x(n) z^{-2n} \sum_{n' \text{ even}} x(n'/2) z^{-n'} = \sum_{n=-\infty}^{+\infty} x^{up}(n) z^{-2n}$$

where

$$x^{up}(n) = \begin{cases} x(n/2) & \text{if } n = 0, 2, 4, \ldots \\ 0 & \text{otherwise} \end{cases}$$

$X(z^2)$ is therefore the z-transform of the upsampled signal. Thus we are able to express a series of upsampling and downsampling operations very easily in the z-domain.

Often 2d z-transforms on an image can be brought about by a series of 1d z-transforms. The ability of the z-transform to easily encode up- and down-sampling means that it is useful in the analysis of systems used for hierarchical decompositions of images for use in image coding and analysis.