Image transforms

Why transform?

- simplify some operations
- compactly represent given image
- enable easy extraction of information from an image

Background

- Inner product : $< x, y > \quad = \sum_j x_j y_j^*$

- Any 2D vector v can be expressed as a linear combination of a basis B:{e_j}

$$v = \sum_j \alpha_j e_j;$$

$$\alpha_j = < v, e_j >$$

- If e_j are unit length and mututally orthogonal, B is an orthonormal basis (ONB)

Background

Identity matrix = $\{\delta[m-n]\}$

$A^H = A^{*T}$ (conjugate transpose)

Real case

A is symmetric : $A = A^T$;

A is orthogonal: $AA^T = I$ or $A^{-1} = A^T$

Complex case

A is Hermitian : $A = A^H$

A is unitary: $AA^H = A^HA = I$ or $A^{-1} = A^H$

A is real and orthogonal → A is unitary (Converse is not true)

Image transforms – decomposition view

- Image transforms are a way to decompose images in terms of basis images
- A given image can be expressed as a weighted combination of basis images
- The weights are found as the inner product of the given image x and the basis image ϕ.

Image decomposition $$x[m,n]=\sum_{l=0}^{N-1}\sum_{k=0}^{M-1}X[k,l]\phi[m,n,k,l]$$ Synthesis

Weights
or transformed image $$X[k,l]=\langle x[m,n],\phi[m,n,k,l]\rangle=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}x[m,n]\phi^*[m,n,k,l]$$ Analysis

Fixed basis: Fourier, DCT, Slant Haar, Hadamard, Transforms

Non-fixed basis: Wavelet

Image transforms – matrix view

Any image x can be transformed to X by pre- and post-multiplying x with matrices

$X = \mathrm{P}x\mathrm{Q}$ analysis

$\therefore x = \mathrm{P}^{-1}X\mathrm{Q}^{-1}$ synthesis

We can also write these using double summations:

$$X[k,l] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} P[k,m]x[m,n]Q[n,l]$$

$$x[m,n] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} P^{-1}[m,k]X[k,l]Q^{-1}[l,n]$$

Choice of P and Q matrices

P and Q are need to be chosen such that

- inversion of the transformation is possible
  - $x \Leftrightarrow X$
- transform (X) computation is efficient
- X reveals some information about x

Types of transforms

- If ϕ (m,n,k,l) = ϕ_1(m,k) ϕ_2(n,l) then we have a separable transform

- If ϕ (m,n,k,l) is a set of orthogonal basis then we have a orthogonal transform

$$< \phi[m,n,k,l], \phi^*[m,n,k',l'] > \;=\; \delta(k-k', l-l')$$

- If ϕ (m,n,k,l) is a complete set then

$$\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \phi[m,n,k,l]\phi^*[m',n',k,l] = \delta(m-m', n-n')$$

- If all the above hold, P=Q and we have a separable, unitary transform

X(k,l) = PxP^T and x(m,n) = ?

Discrete Fourier Transform

DFT: x → X; with a ONB made of complex sinusoids

- Elements of P and Q matrices are complex exponentials

  If x is of size MxN, then P is MxM and Q is NxN

$$P[k,m] = \frac{1}{M} e^{-j\frac{2\pi}{M}km} \qquad Q[n,l] = \frac{1}{N} e^{-j\frac{2\pi}{N}nl}$$

- X is a complex valued function
  - Two images are needed to display X: {amplitude and phase} or {real and imaginary}

Note: P = Q

Discrete Fourier Transform

Let P = Q = $F(k,m) = e^{-j\frac{2\pi}{N}mk} = W_N^{-mk}$

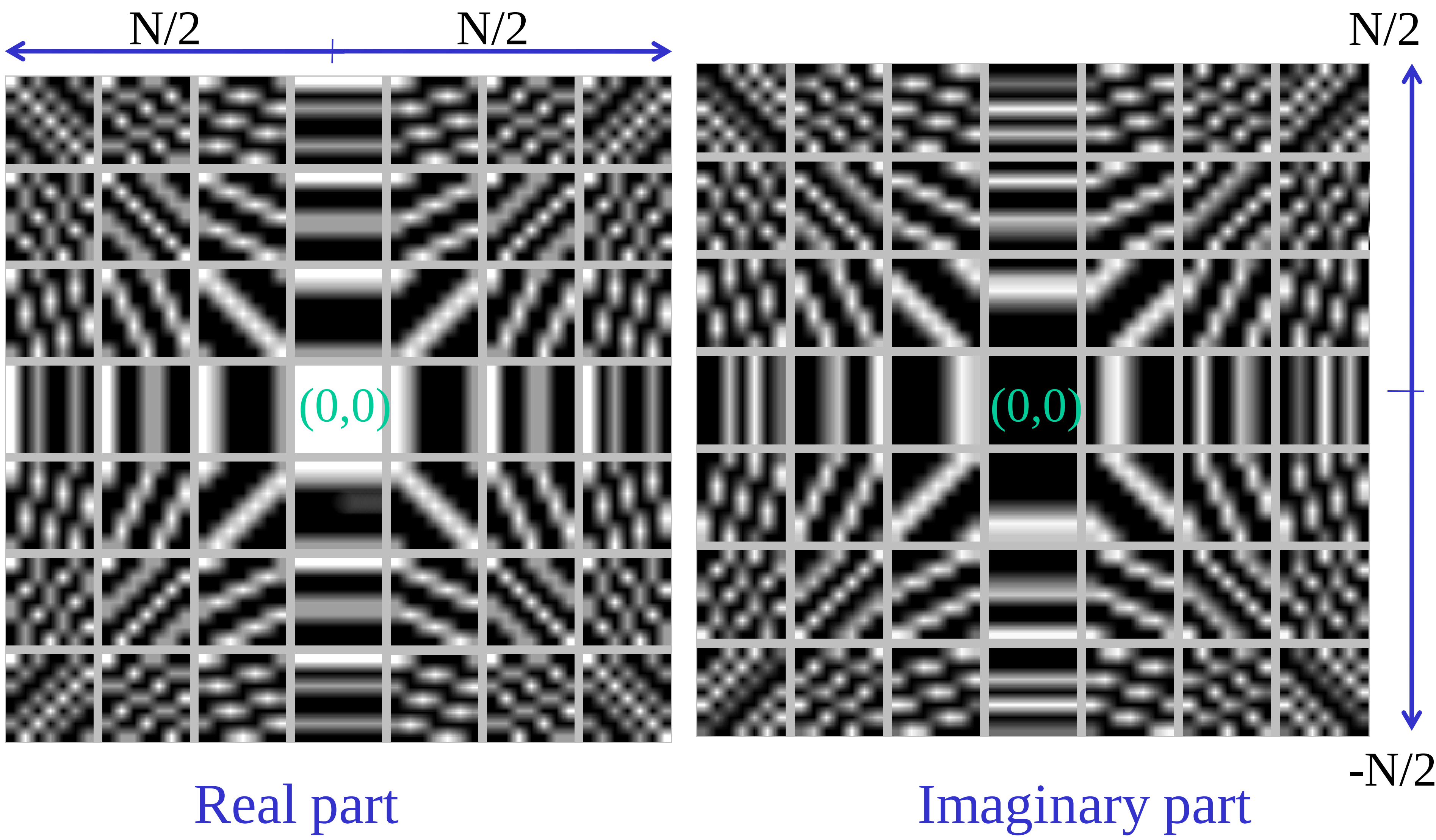
Then X = F x F and x = F^{-1} X F^{-1}

- F is unitary <u>and</u> symmetric
  - $F^{-1} = F^H$ and $F = F^T$ ➔ $F^{-1} = F^*$

- Basis images for a NxN image :

$$\{\frac{1}{N} W_N^{(mk+nl)}\} \qquad 0 \le k,l,m,n \le N-1$$

Basis Images of DFT



Symmetry properties of DFT

- The basis function is periodic (with N)

$$X(k) = X(k \pm N)$$

$$\boxed{X(k+aN, l+bN) = X(k,l) \quad 0 \le k, l, \le N-1}$$

- Conjugate symmetry about N/2

$$X(\frac{N}{2} - k) = X^*(\frac{N}{2} + k)$$

$$\boxed{\begin{gathered} X(\frac{N}{2} \pm k, \frac{N}{2} \pm l) = X^*(\frac{N}{2} \mp k, \frac{N}{2} \mp l) \\ X(k,l) = X^*(N-k, N-l) \end{gathered}}$$

$$0 \le k, l, \le \frac{N}{2} - 1$$

➢ There are only *N* Unique values for X

DFT computation

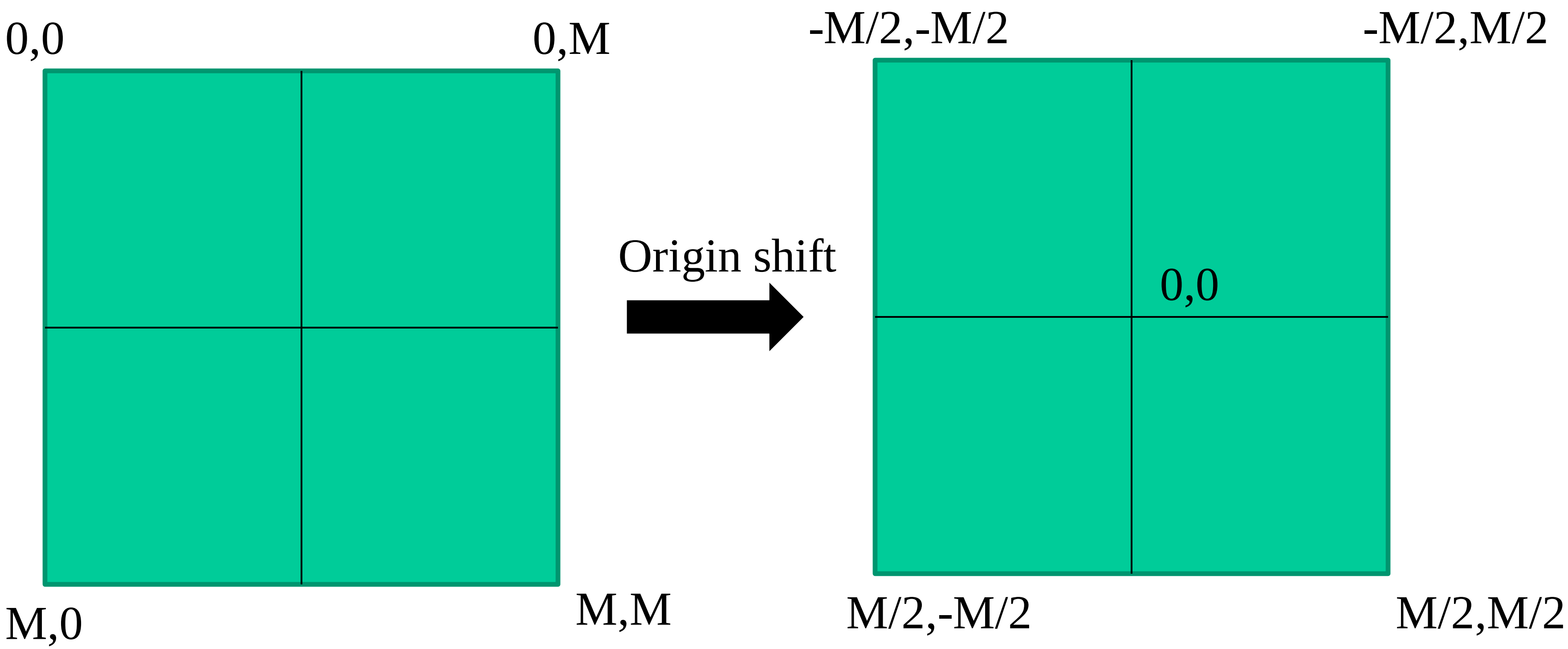
- When one computes a NxN DFT for an image x, the result does not capture 1 complete period of the spectrum if origin is set at top left corner

- Hence, to display **one entire period of the spectrum** we **shift the origin to the centre**

- To do this we note $$(-1)^{m+n} x(m,n) \Leftrightarrow X(k-\frac{N}{2}, l-\frac{N}{2})$$

Checker board pattern

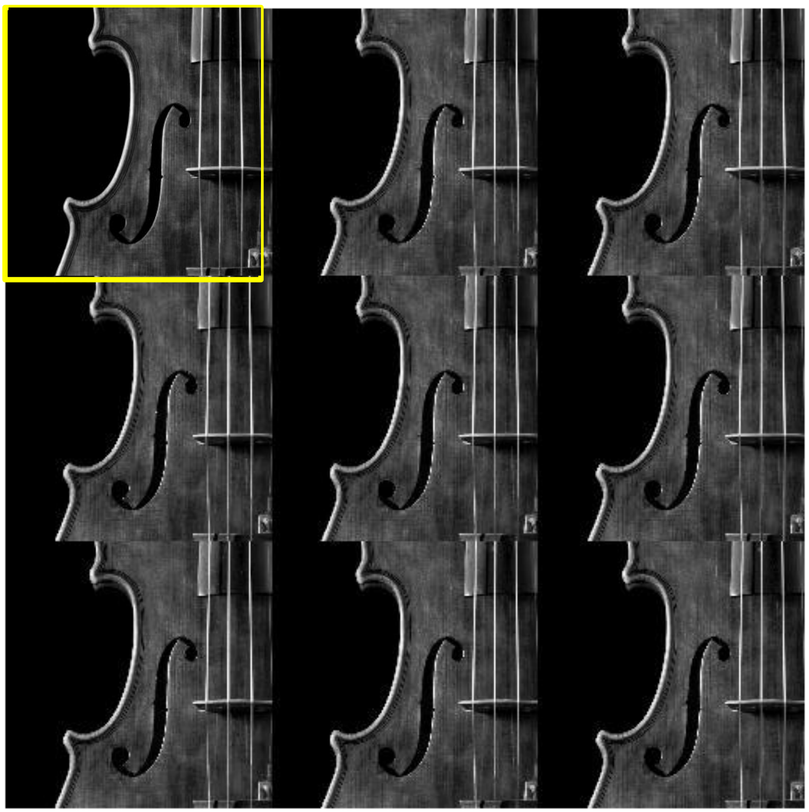
Computing and displayingX[k,l]

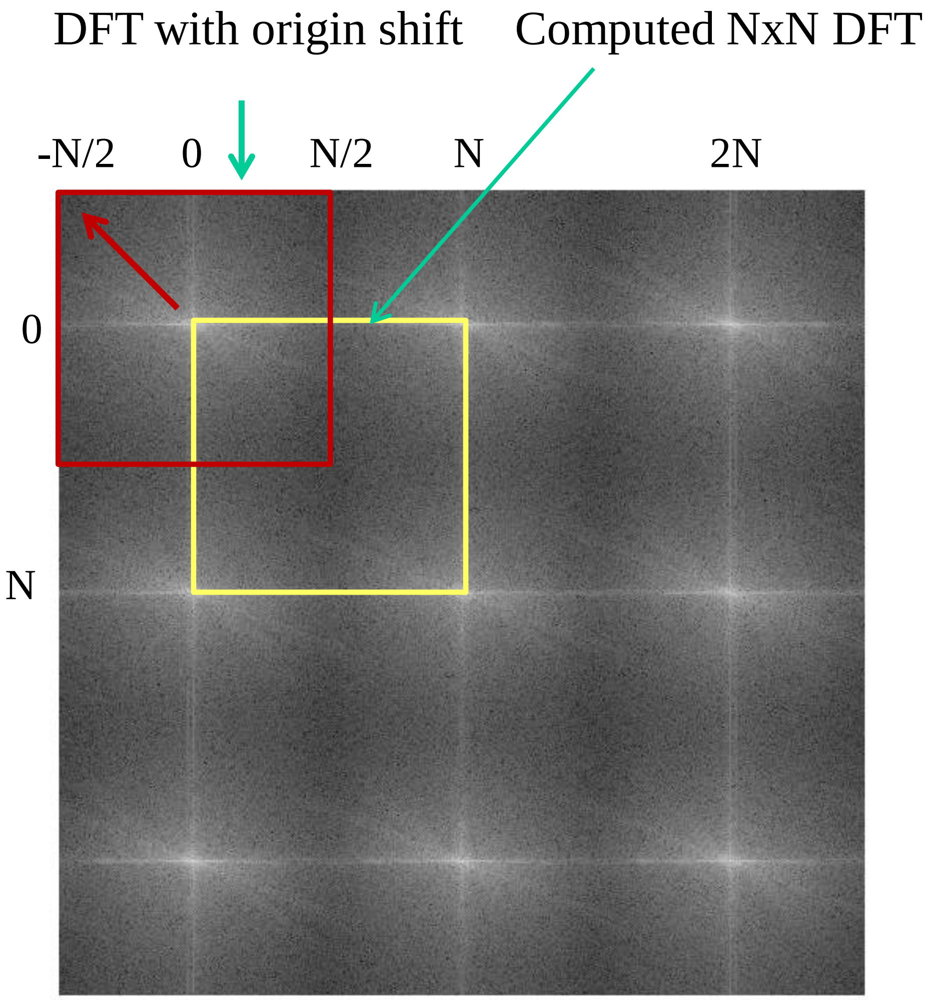


Computed DFT
Does not help visualise 1-full period of the spectrum

Displayed DFT
Covers 1-full period of the spectrum

DFT interpretation

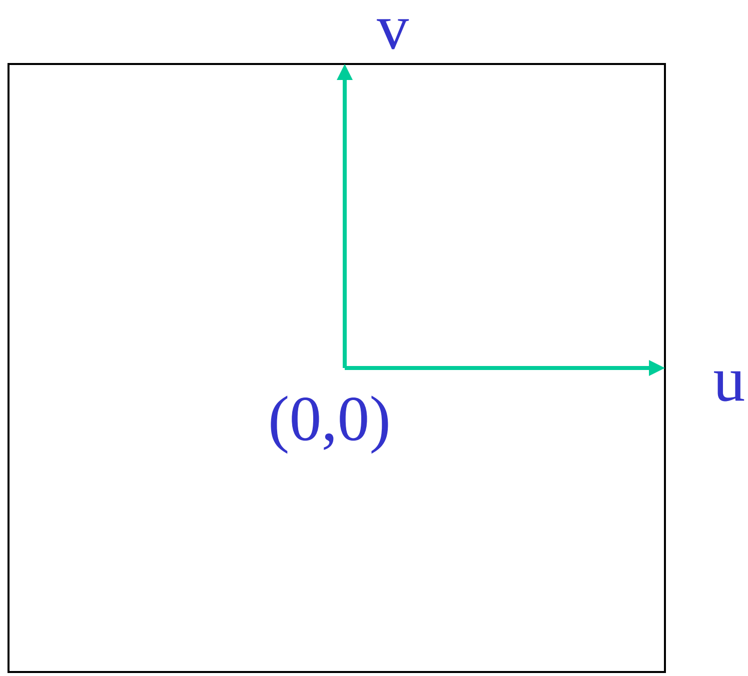




Violin image → DFT (amplitude) spectrum

Violin image is periodically extended! ← It is periodic!

DFT spectra of images



Transform

|X|: Magnitude spectrum

∠X: Phase spectrum

Amplitude spectrum

The amplitude spectrum of an image typically has

- A very high dynamic range
  - Well above the 256 levels of a 8-bit display

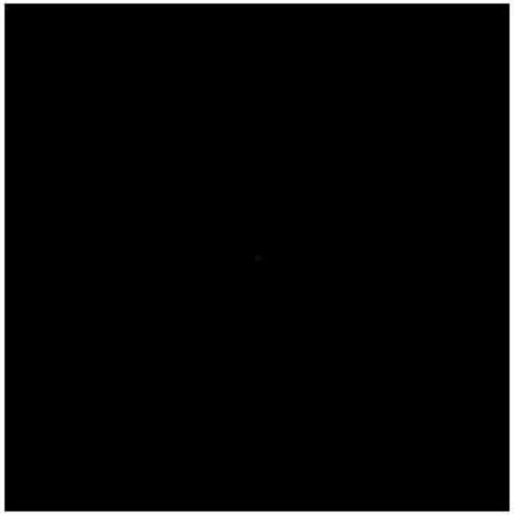
- Energy in LF band > energy in MF band > energy in HF band

- A non linear greyscale transformation is needed to display properly
  - Ex. $Log(1 + |F|)$; F is the Fourier transform of image f

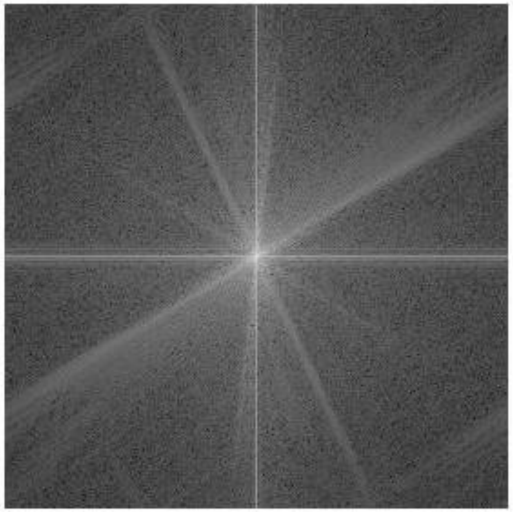
Check: why log transform?

Importance of non-linear stretching

Piano



|Piano|



log(|Piano|+0.01)

DFT examples …

Raw Image

Fourier Amplitude

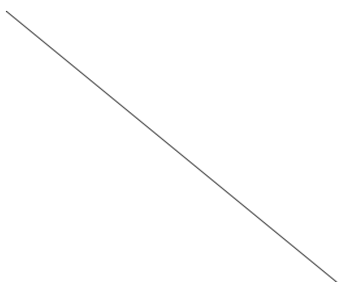
Horizontal Line

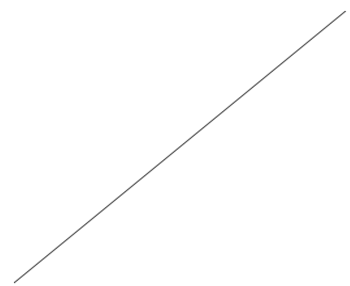
Vertical Line

Vertical Line

Horizontal Line

Diagonal Line



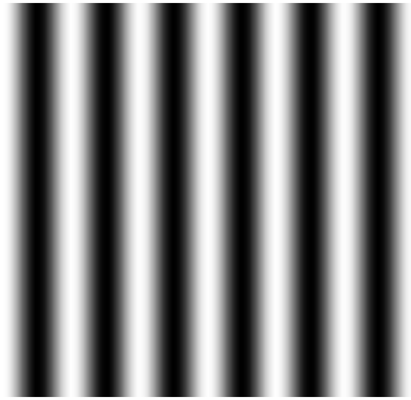


Diagonal Line

DFT examples

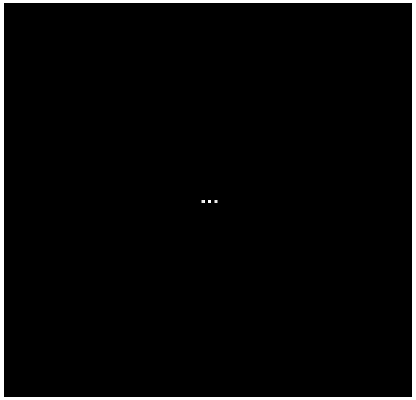
Raw Image | **Fourier Amplitude**

Vertical sinusoid, higher frequency



DC term + side lobes wide spacing

Vertical sinusoid, lower frequency



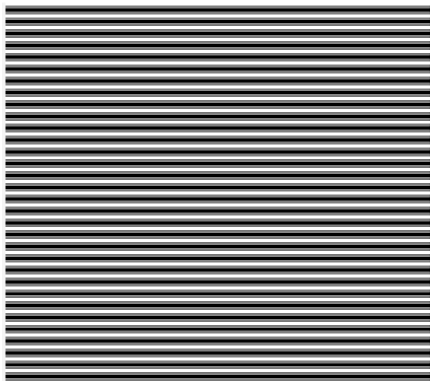
DC term+ side lobes close spacing

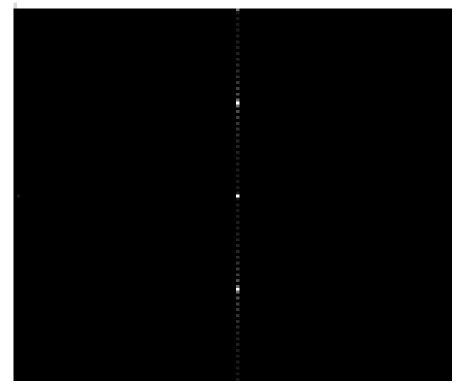
Images from Steve Lehar http://cns-alumni.bu.edu/~slehar An Intuitive Explanation of Fourier Theory

DFT examples …

Raw Image | **Fourier Amplitude**

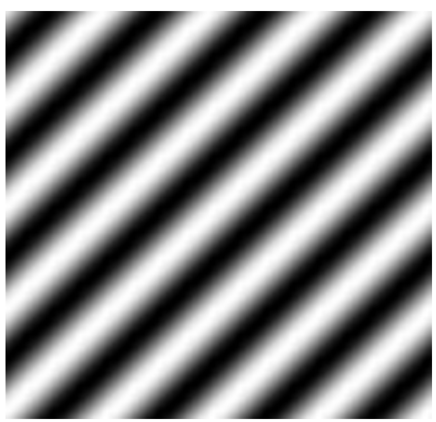
Horizontal sinusoid, higher frequency

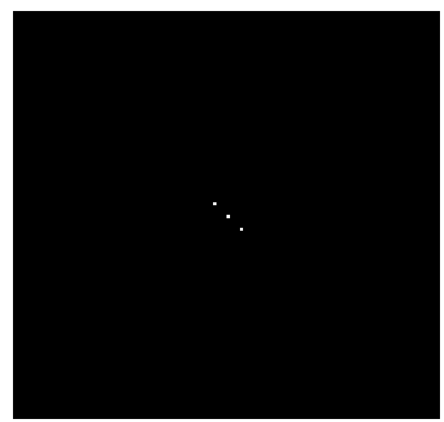




DC term + side lobes wide spacing

Sinusoid, tilted





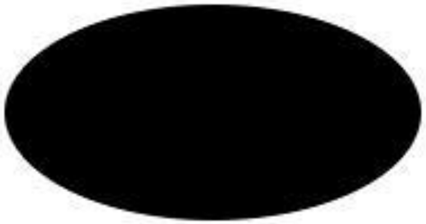
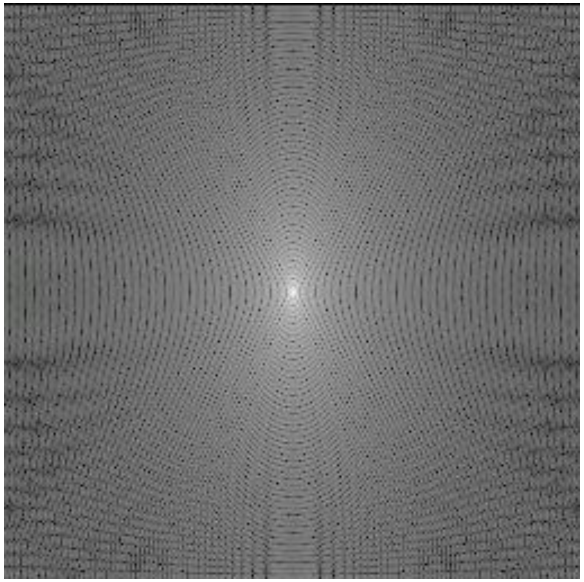
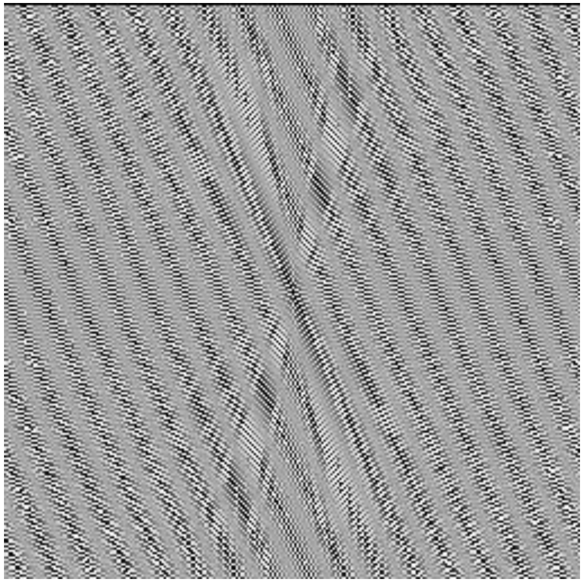
Tilted spectrum

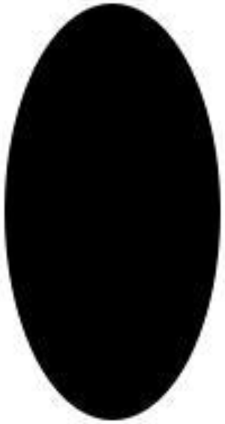
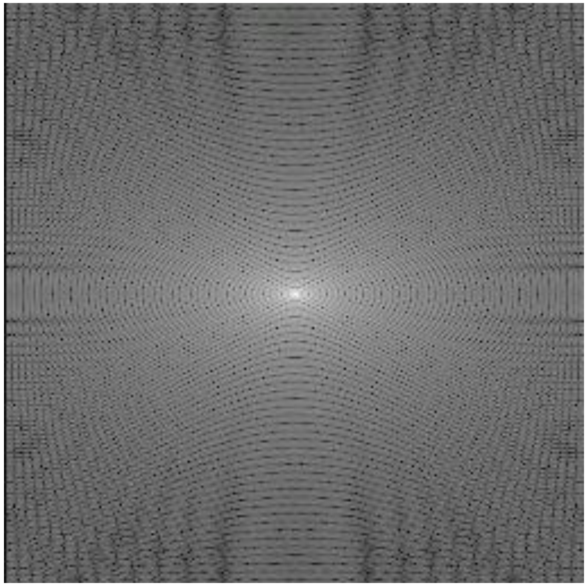
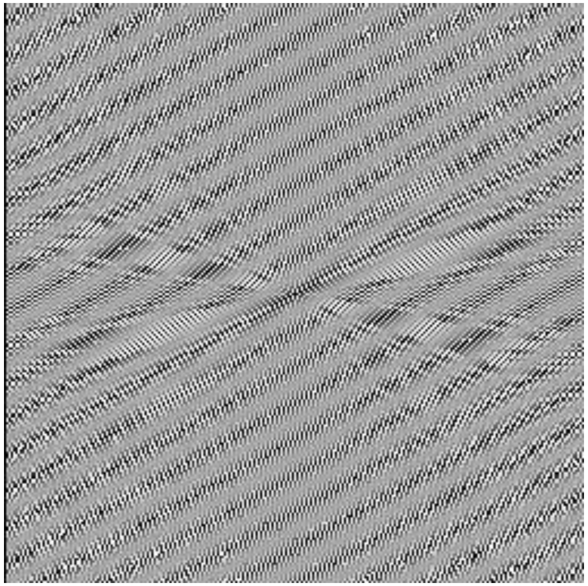
Images from Steve Lehar http://cns-alumni.bu.edu/~slehar An Intuitive Explanation of Fourier Theory

DFT examples: (symmetric & binary) ellipse

|ellipse| ∠ ellipse

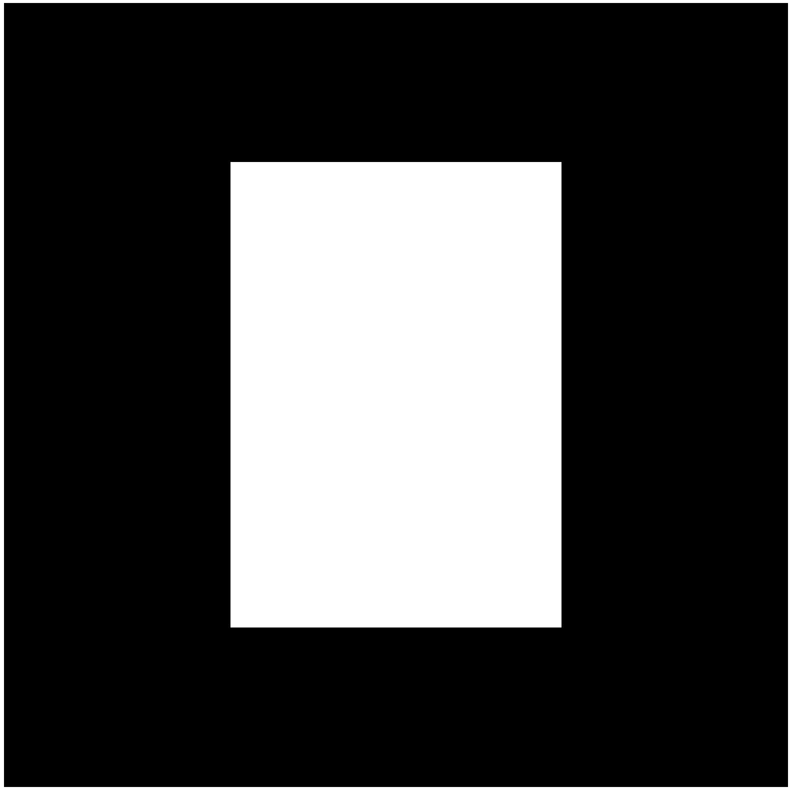
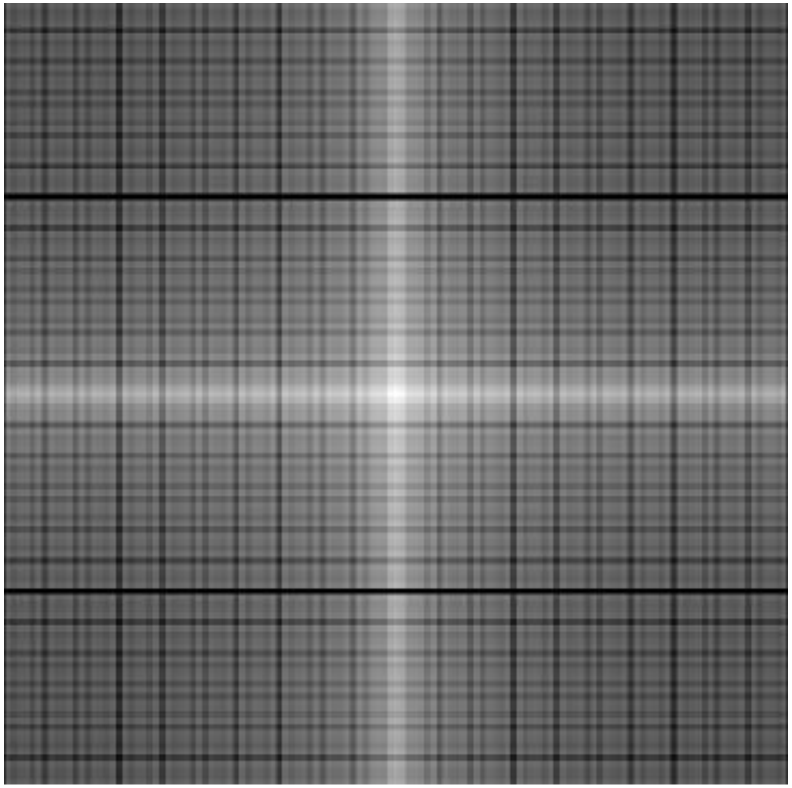
ellipse

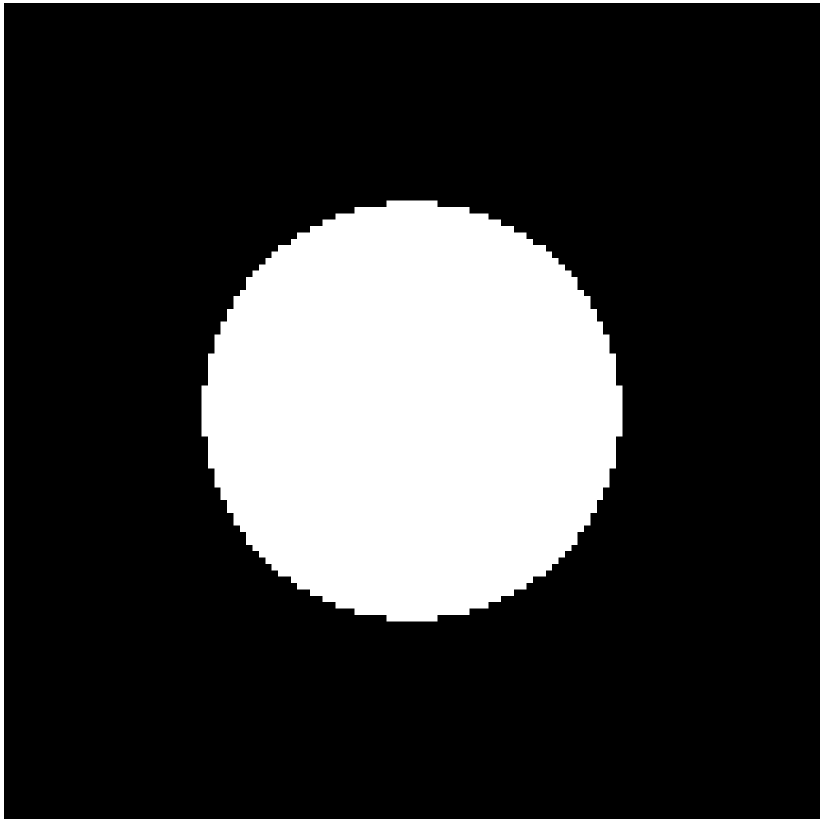
Rotated ellipse

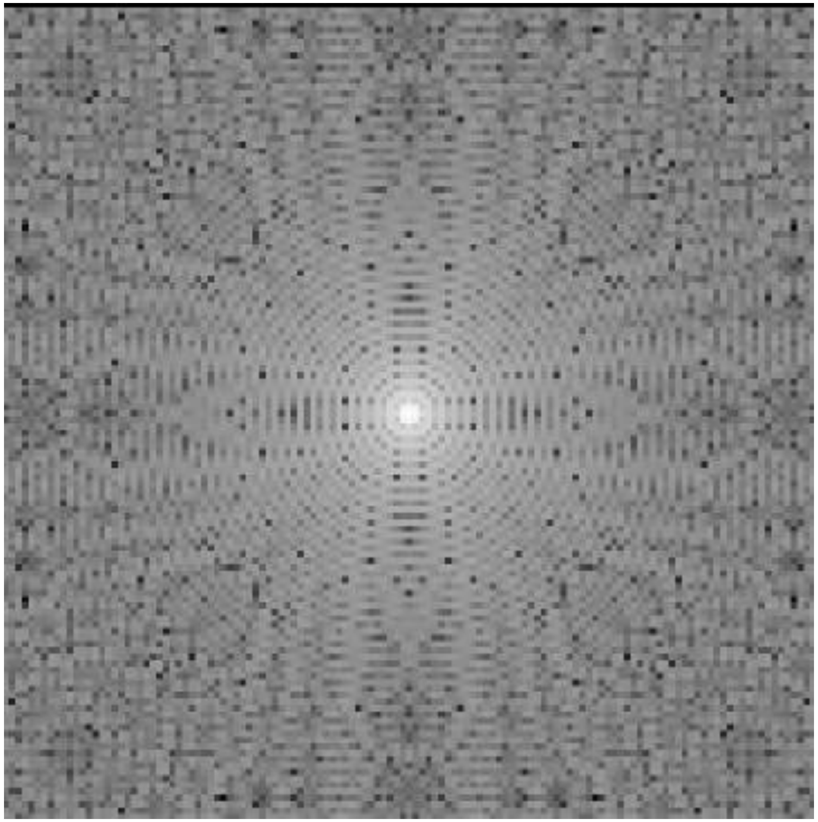
Amplitude spectrum examples: (binary) rectangle

A rectangle and its |FT|

Amplitude spectrum examples: (binary) circle

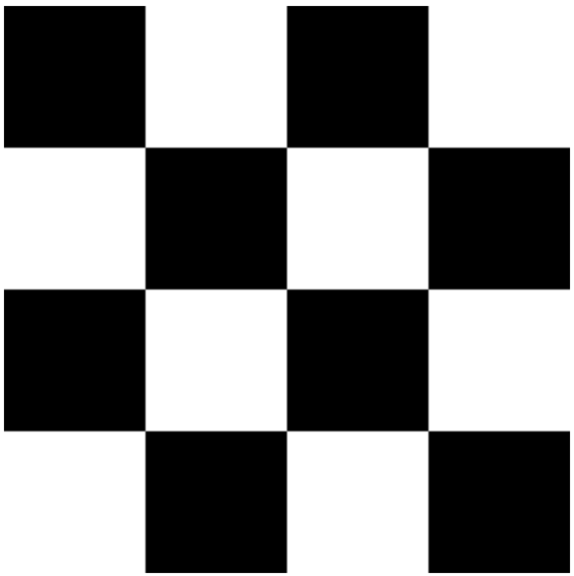


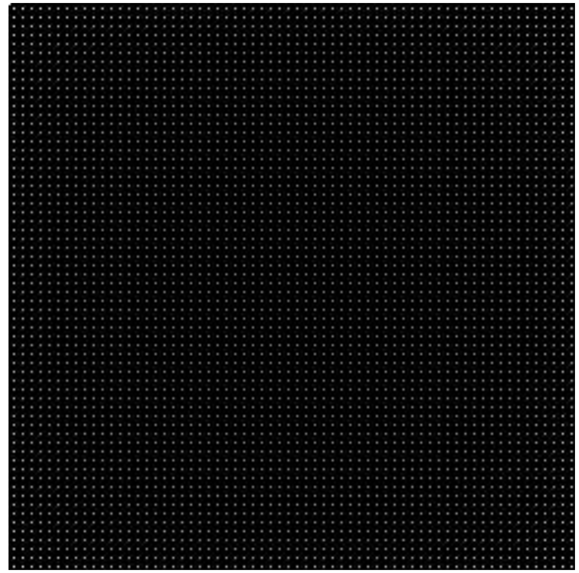


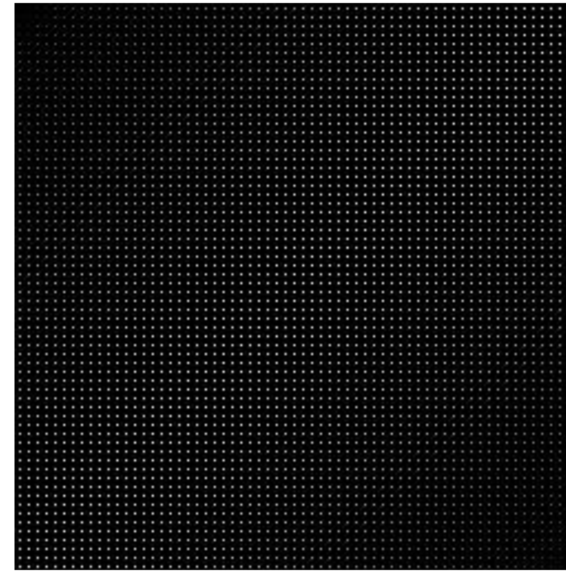
A circle and its |FT|

DFT of a periodic image

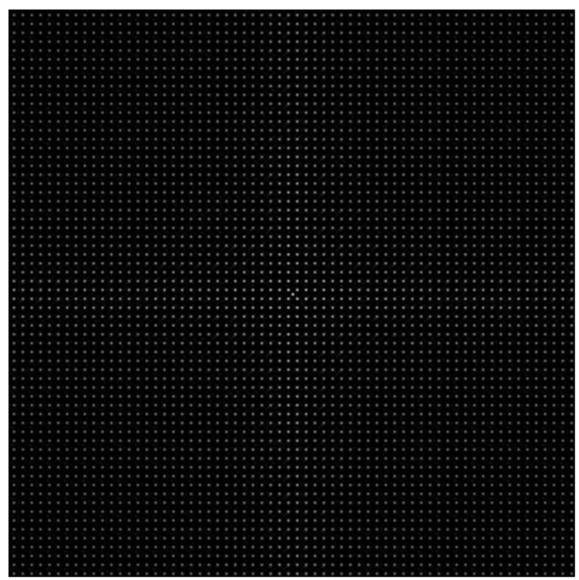
|chess| origin is at top left ∠ chess

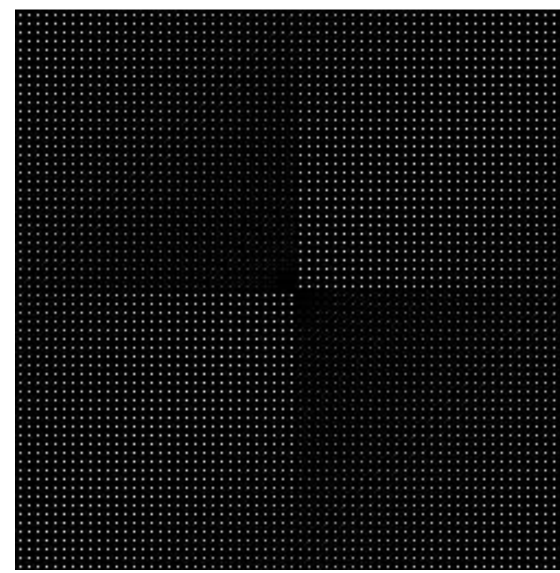






with origin shifted to the centre





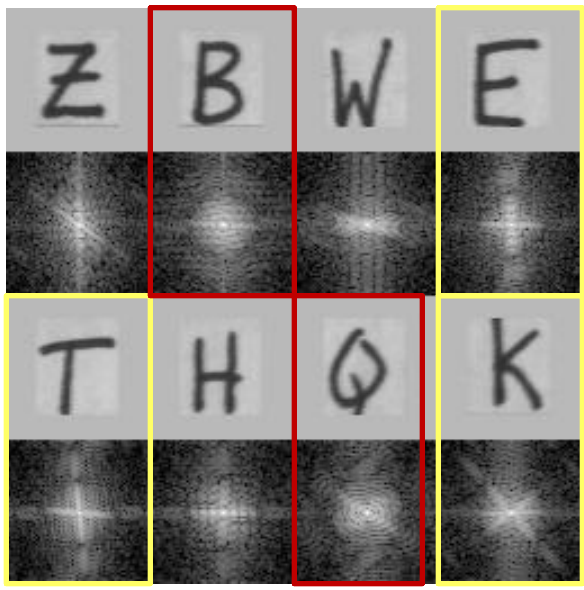
DFT of greyscale images - effect of strong edges

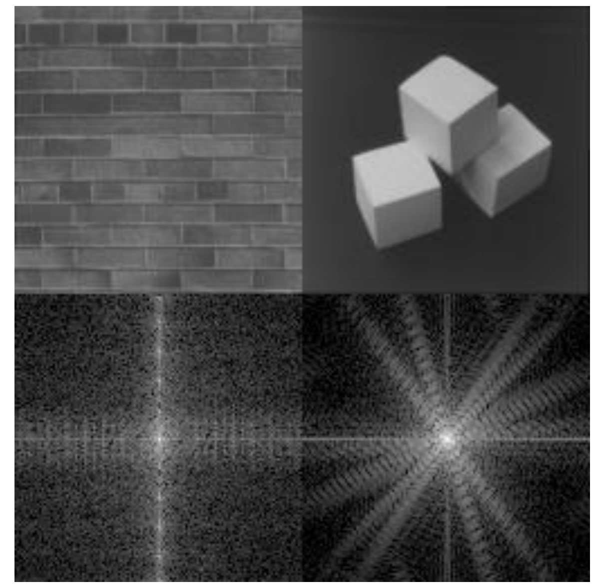
Images

Amp. spectra

Images

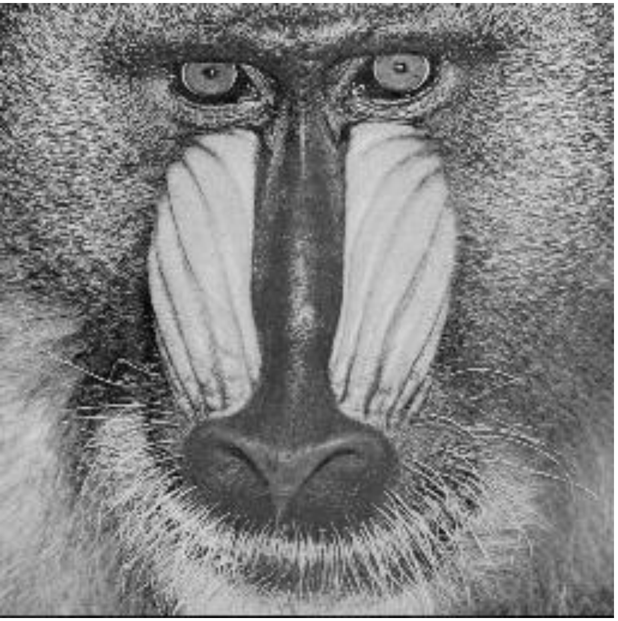
Amp. spectra

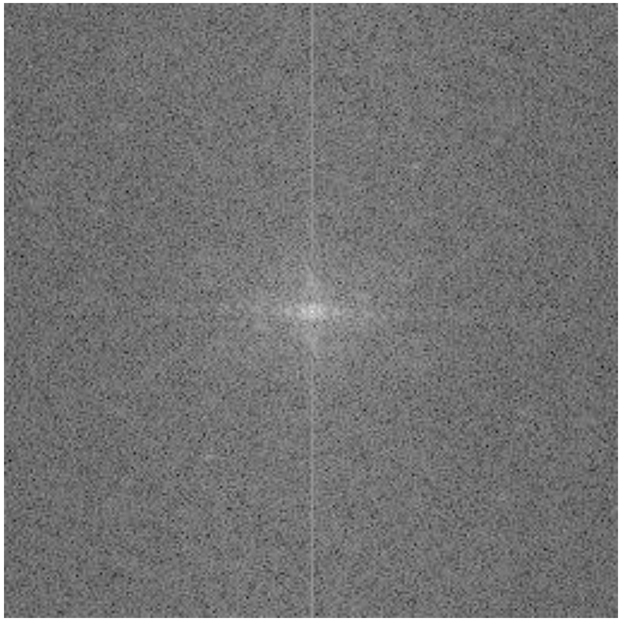


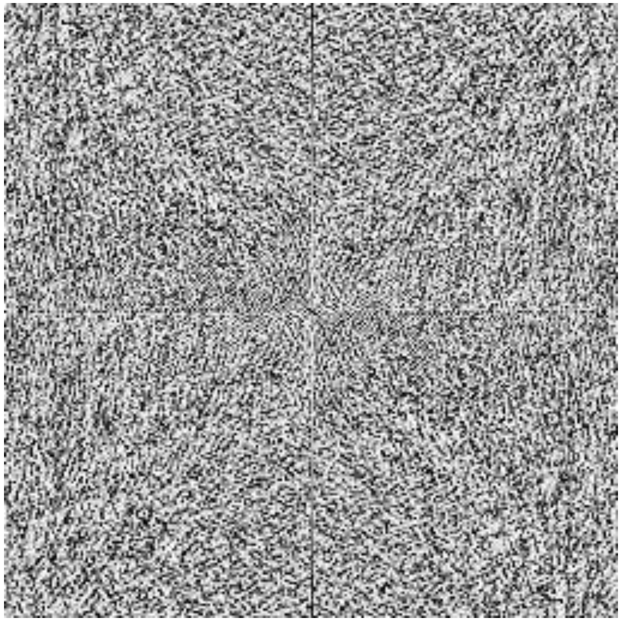


http://www.cs.unm.edu/~brayer/vision/fourier.html

DFT examples: (greyscale image) Ape







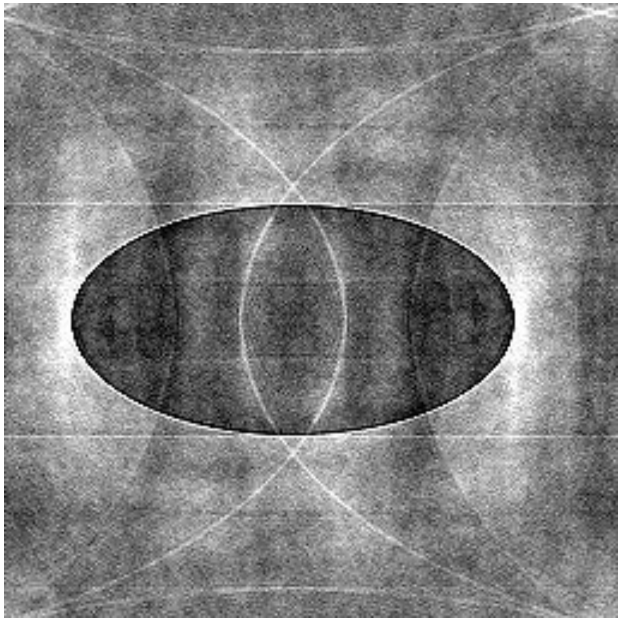
Ape

|ape|

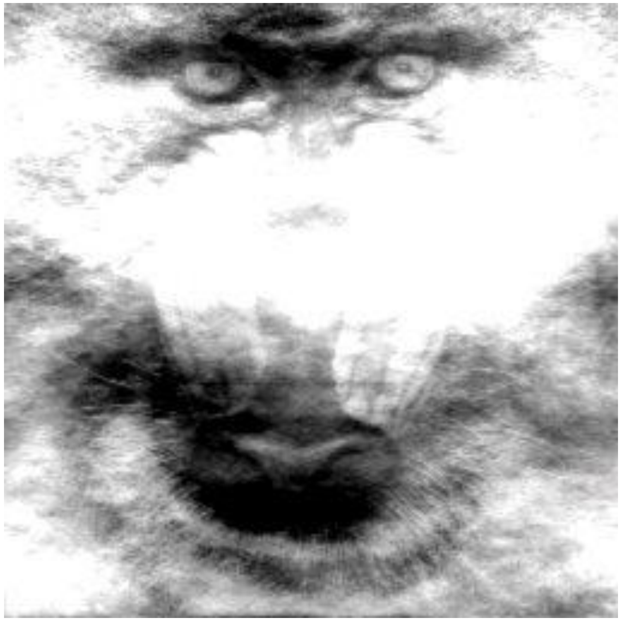
∠ ape

Inverse DFT with incorrect phase spectra

IDFT {|ape|, ∠ellipse}



IDFT {|ellipse|, ∠ ape}



The phase spectrum appears to be strongly encoding the shape

Filtering in the Transform domain

Linear Filtering

- Given *x[m,n]* the filtered image *y[m,n]* is

$$y[m,n] = (x * h)[m,n]$$
$$\Downarrow$$
$$y[m,n] = \Im^{-1}\{Y\} = \Im^{-1}\{X[k,l]H[k,l]\}$$

where *H[k,l] = DFT{h[m,n]}*

- *H* can be viewed as a frequency domain **mask**
- Mask is *binary valued* for *ideal* filtering and *grey valued* for *non-ideal* filters (ex. Gaussian)
- Size of H and X are equal
  - Y is the result of multiplying X and H

Convolution ⇔ product

An important property of Fourier transform:

$$\Im\{ f[m,n] * h[m,n] \} = F[k,l]H[k,l] \qquad (1)$$

- * here is a <u>periodic</u> convolution since DFT and IDFT are periodic
- So (1) will result in wraparound errors if we want <u>linear</u> convolution

Implementing <u>linear</u> convolution using DFT:

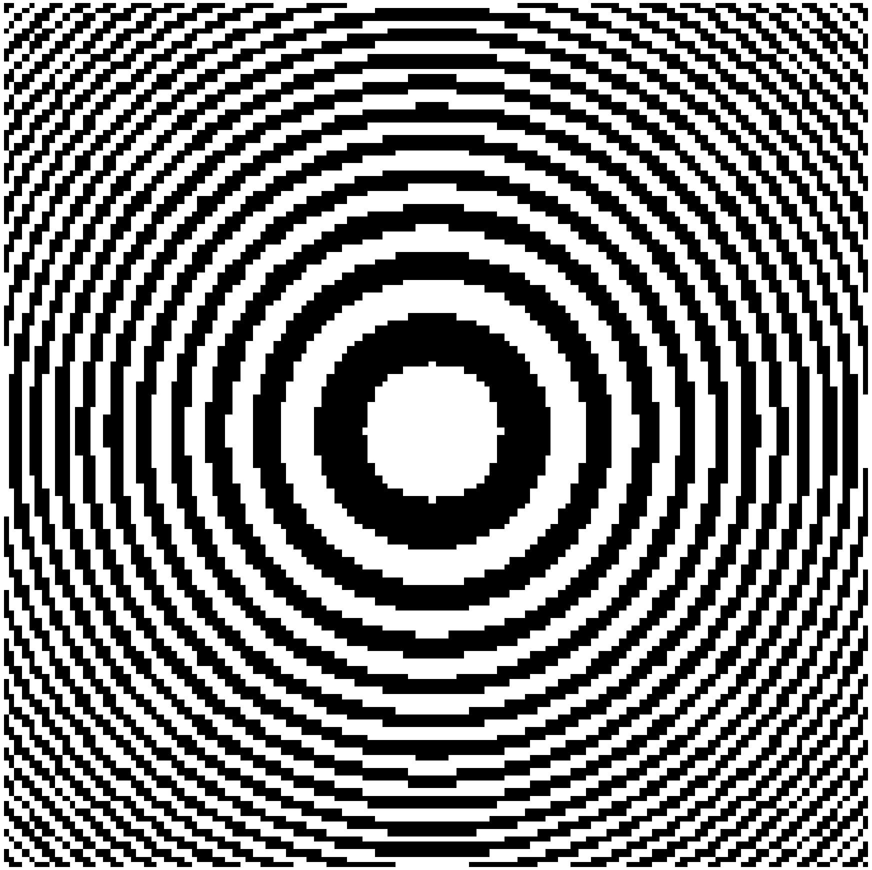
Let f be MxM and h be LxL

$\therefore$ linear convolution of f and h will result in an image of size M+L-1

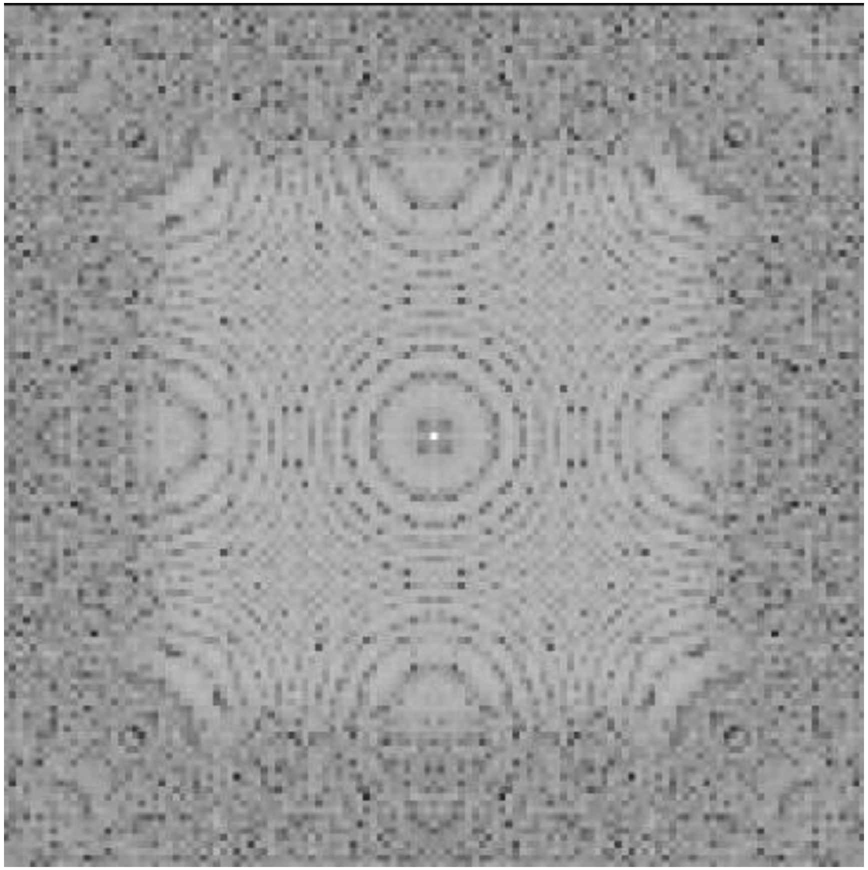
Extend f and h to be (M+L -1) x (M+L-1) by zero-padding

before implementing (1)

Test image1 and its DFT

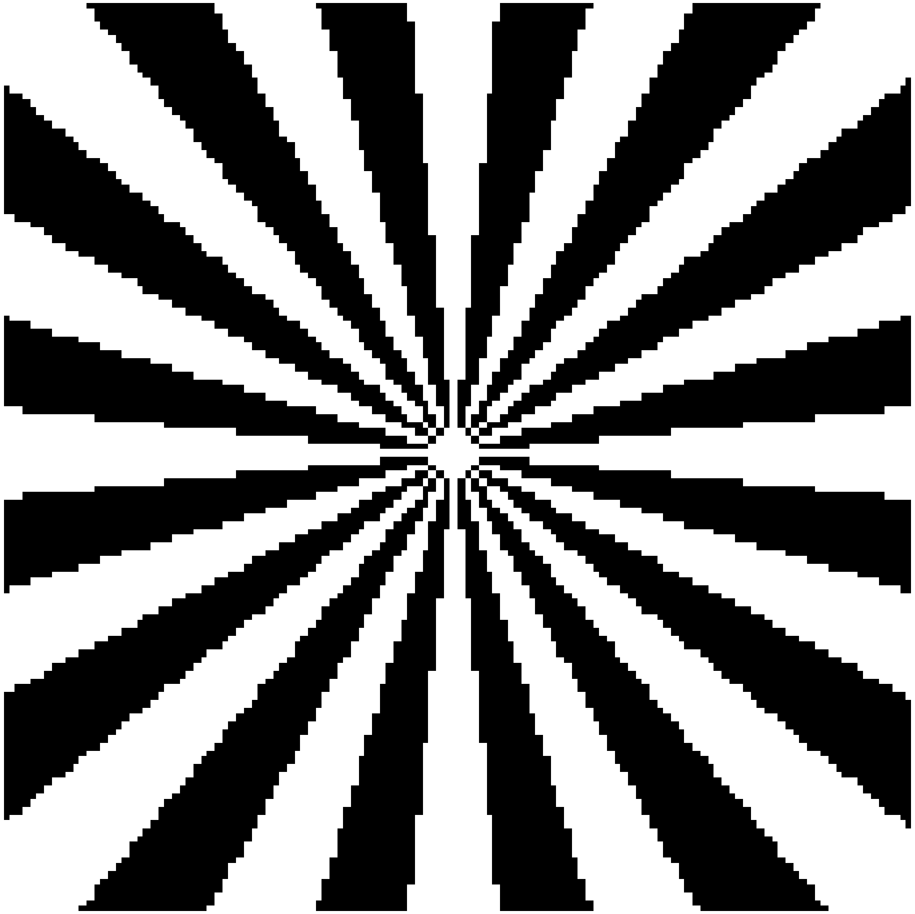


rings

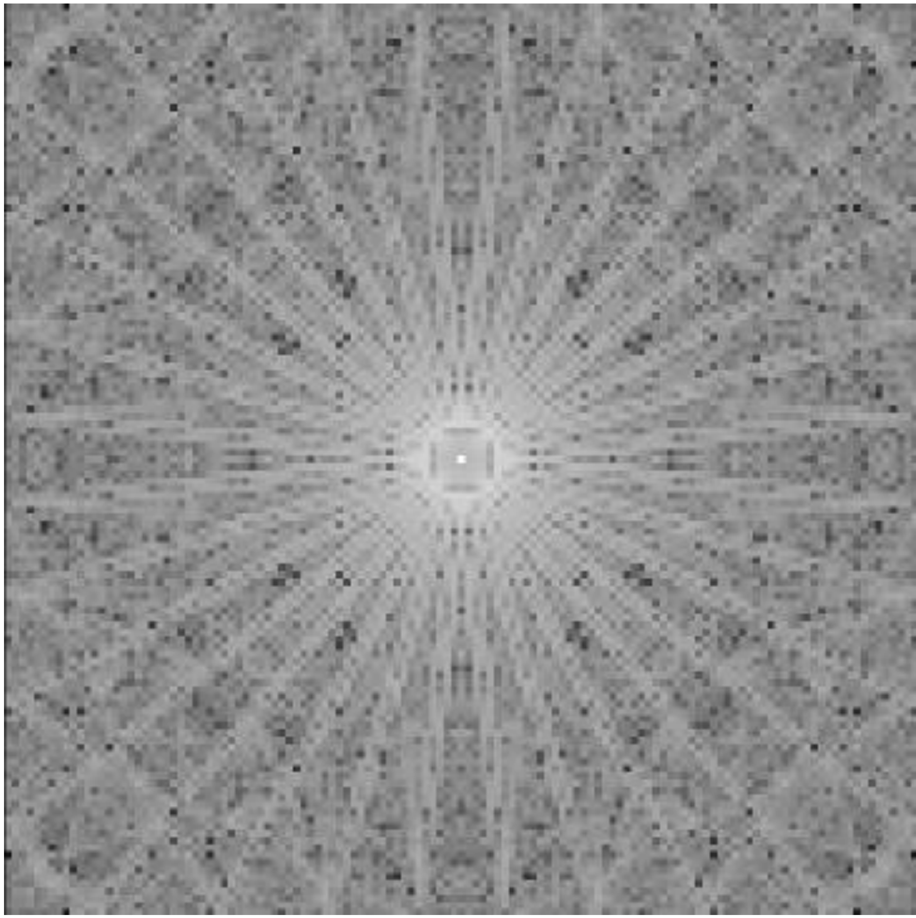


|rings|

Test image2 and its DFT

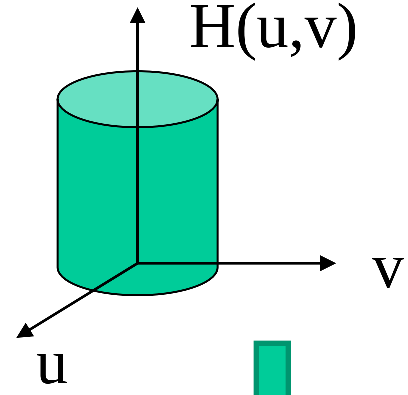


star



|star|

Low pass filtering



- **Ideal/brickwall filter**

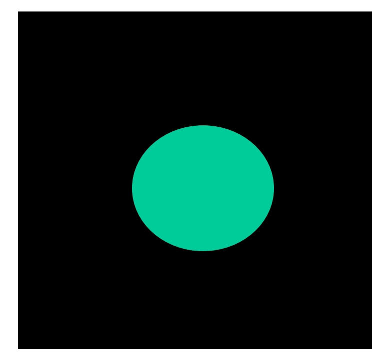
$$H(u,v) = 1\ ;\ \ \omega = \sqrt{u^2 + v^2} < \omega_c = \sqrt{{u_0}^2 + {v_0}^2}$$
$$= 0 \text{ otherwise}$$

ω_c is the cutoff frequency

- **Non ideal filter - Butterworth filter**

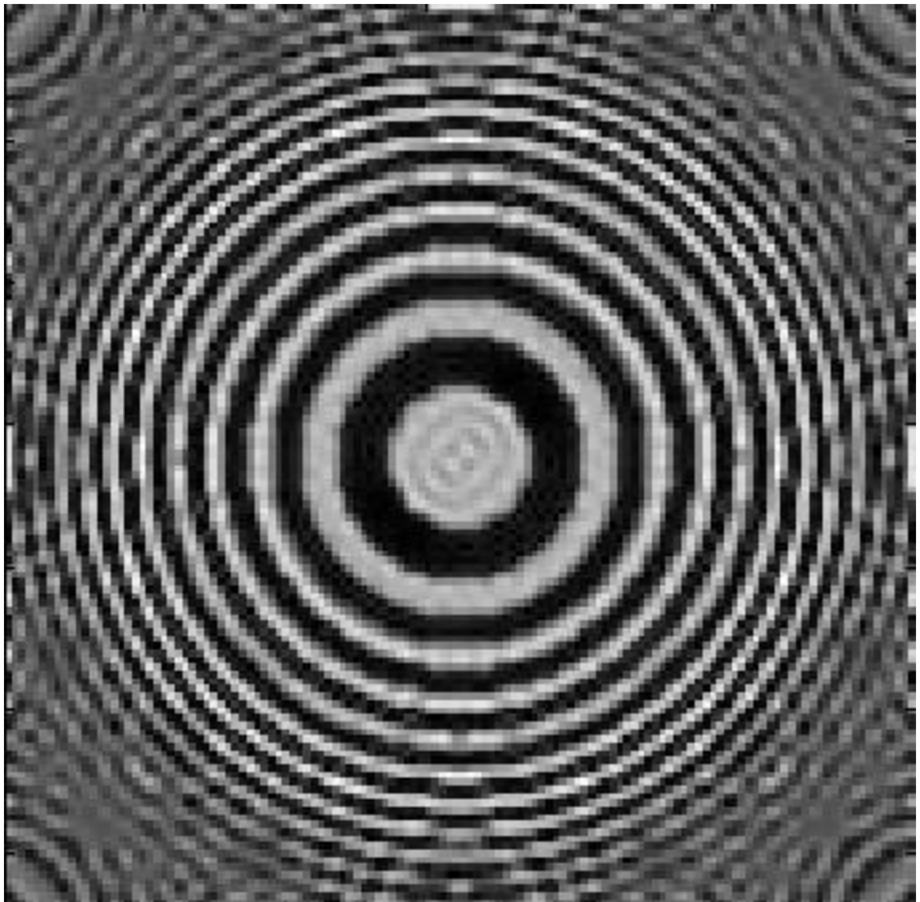
$$H(u,v) = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2n}}}$$

n is the order of filter

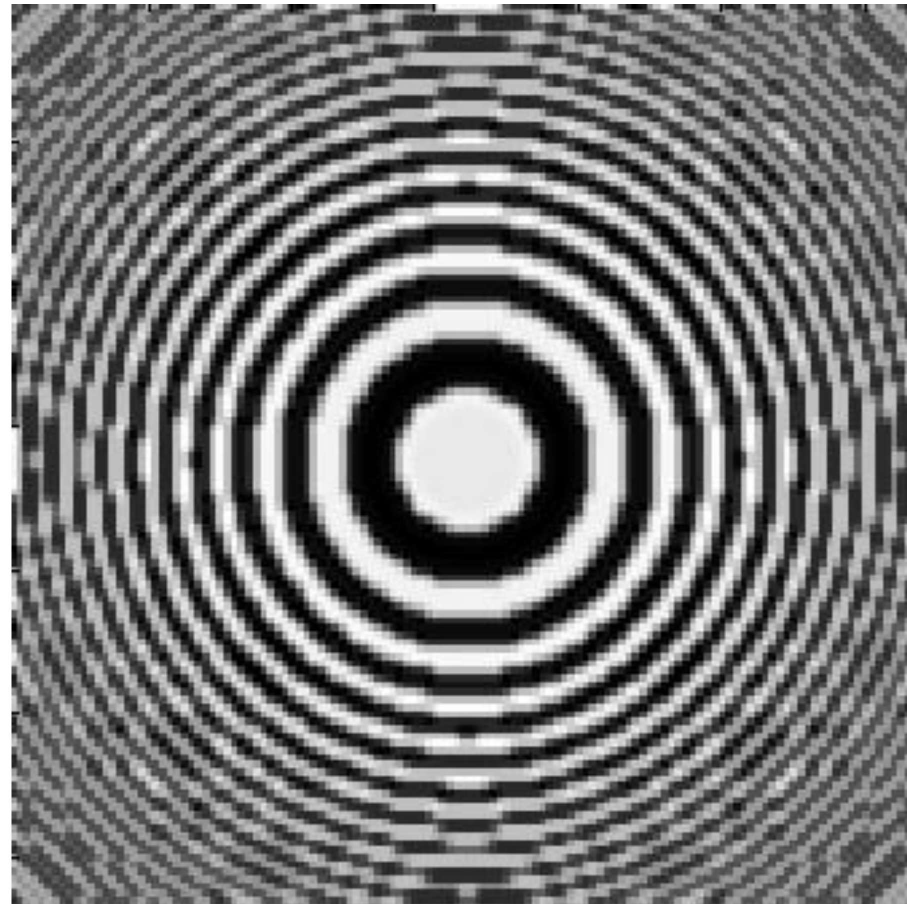


u-v plane binary mask

Low pass filtered rings

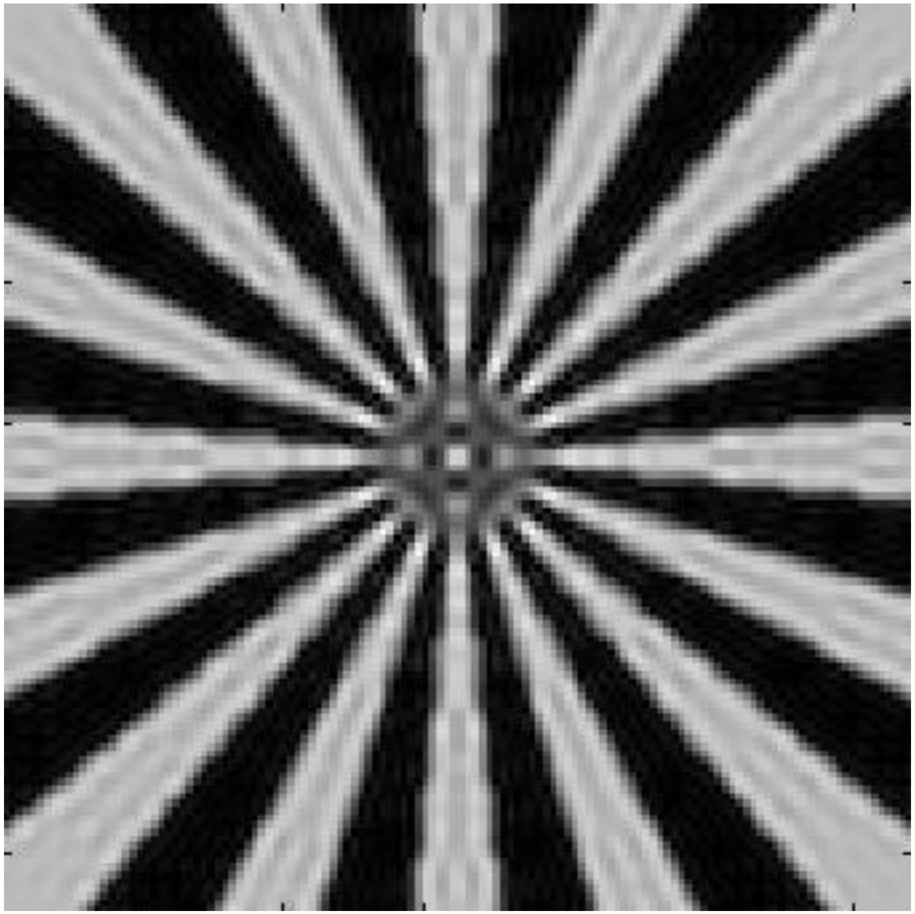


Ideal filter

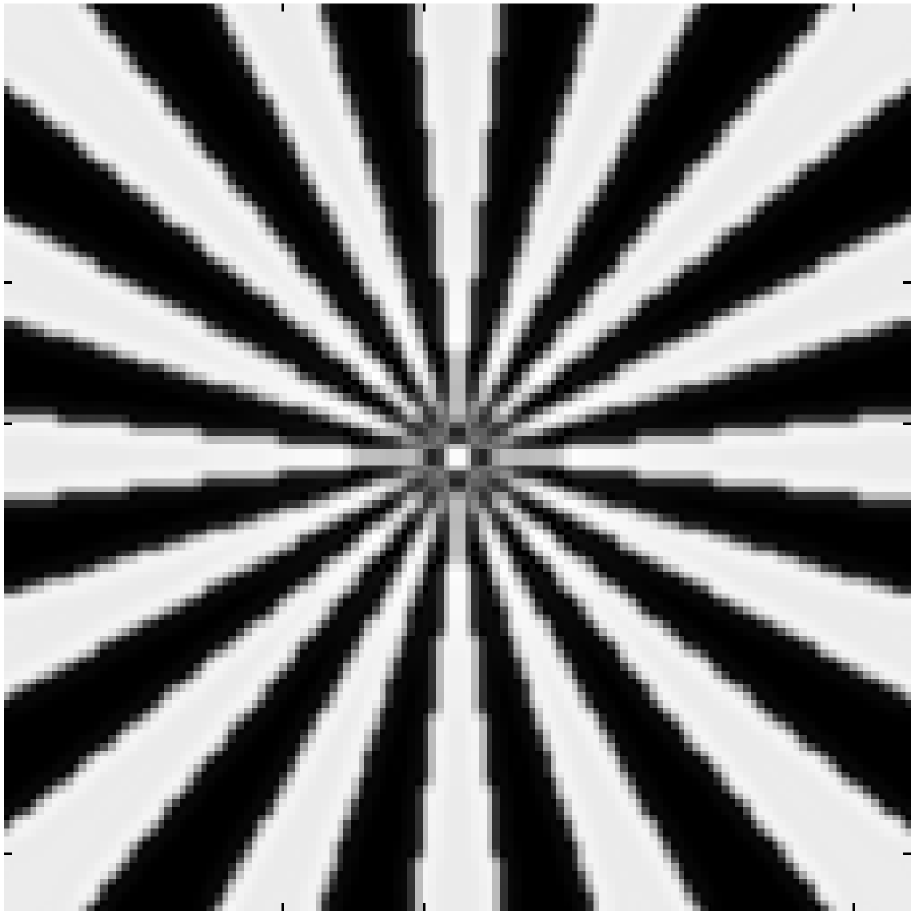


Butterworth filter

Low pass filtered star



Ideal filter

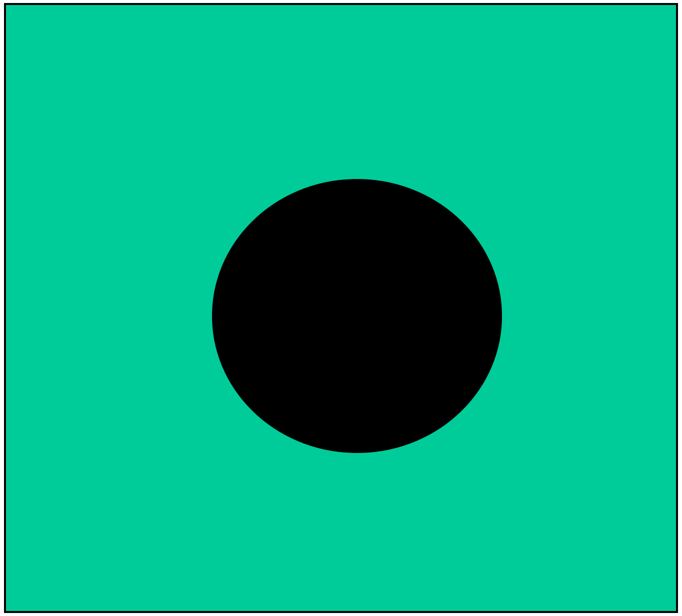


Butterworth filter

High pass filtering

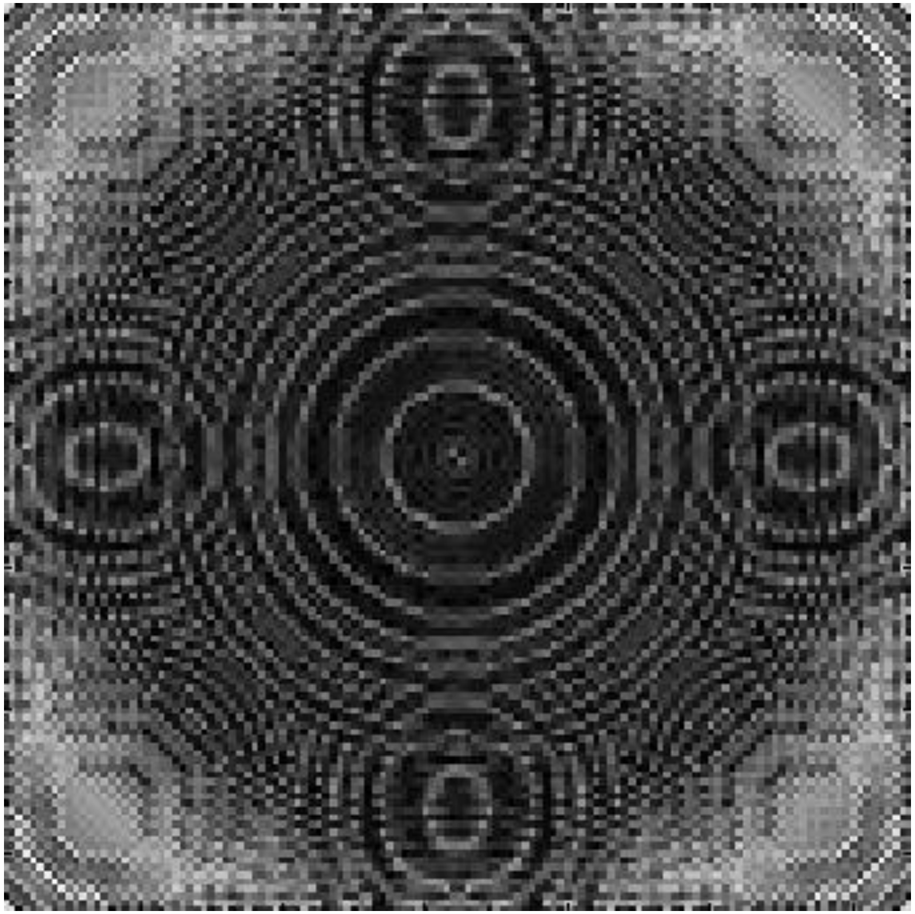
(Ideal) $H(u,v) = 1$; $(u^2+v^2)^2 > (u_0{}^2+v_0{}^2)^2$

$= 0$ otherwise

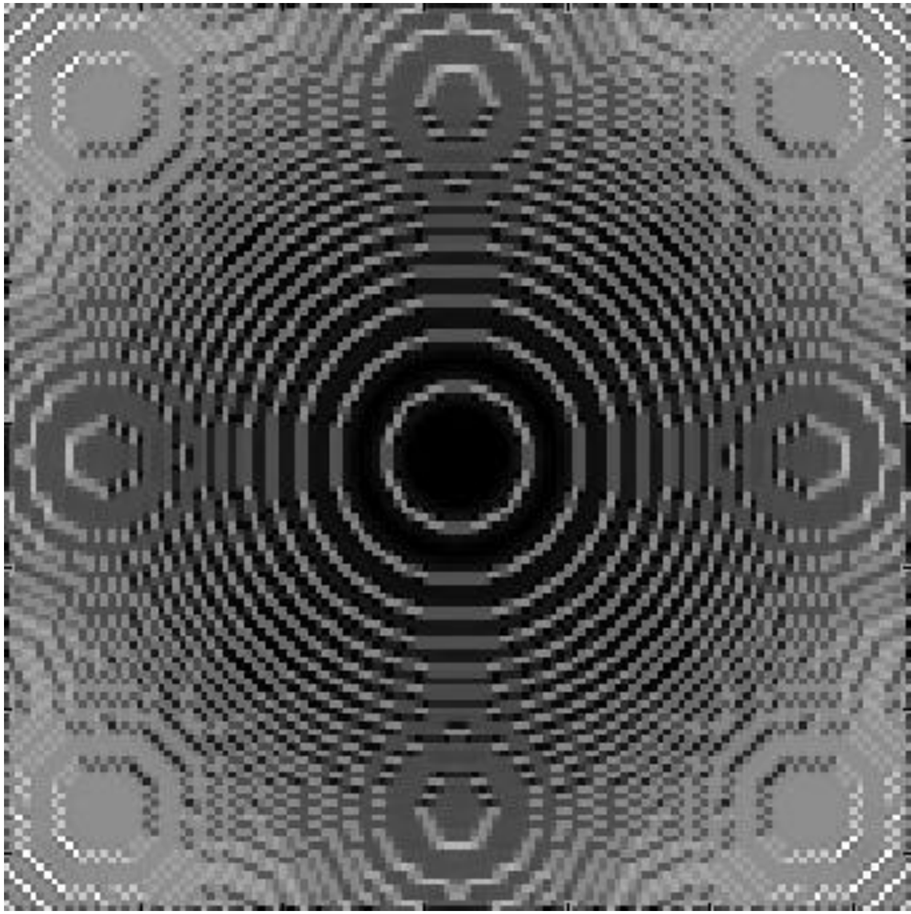


u-v plane binary mask

High pass filtered ring

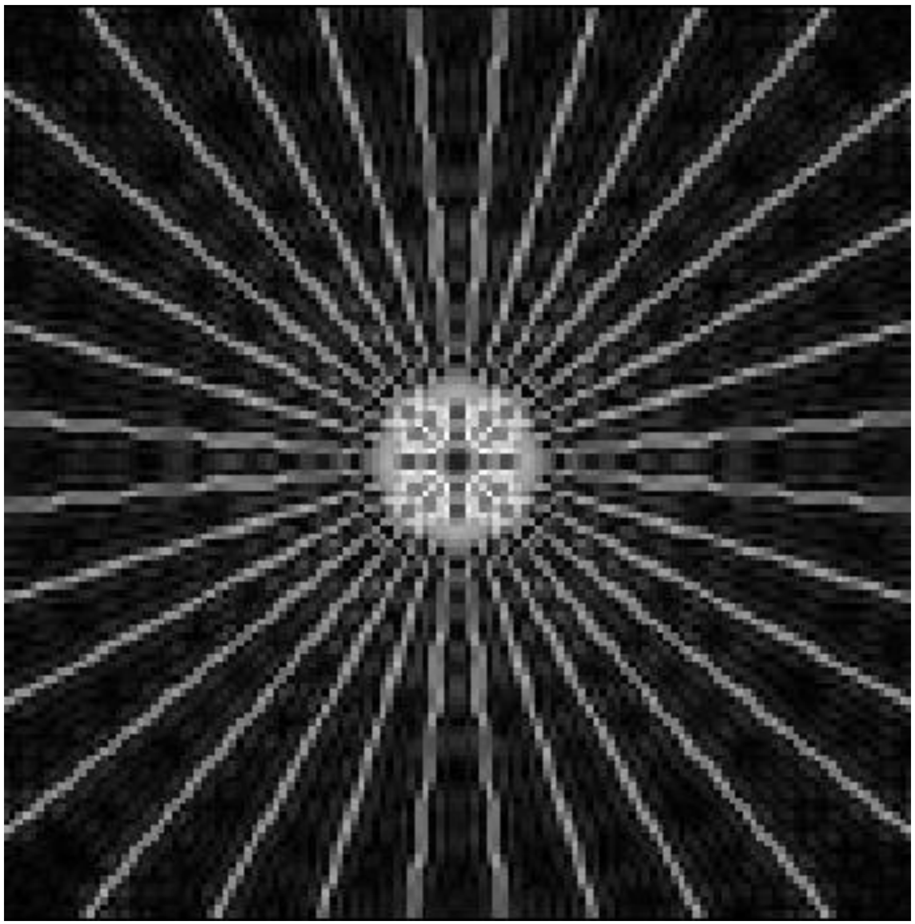


Ideal HPF

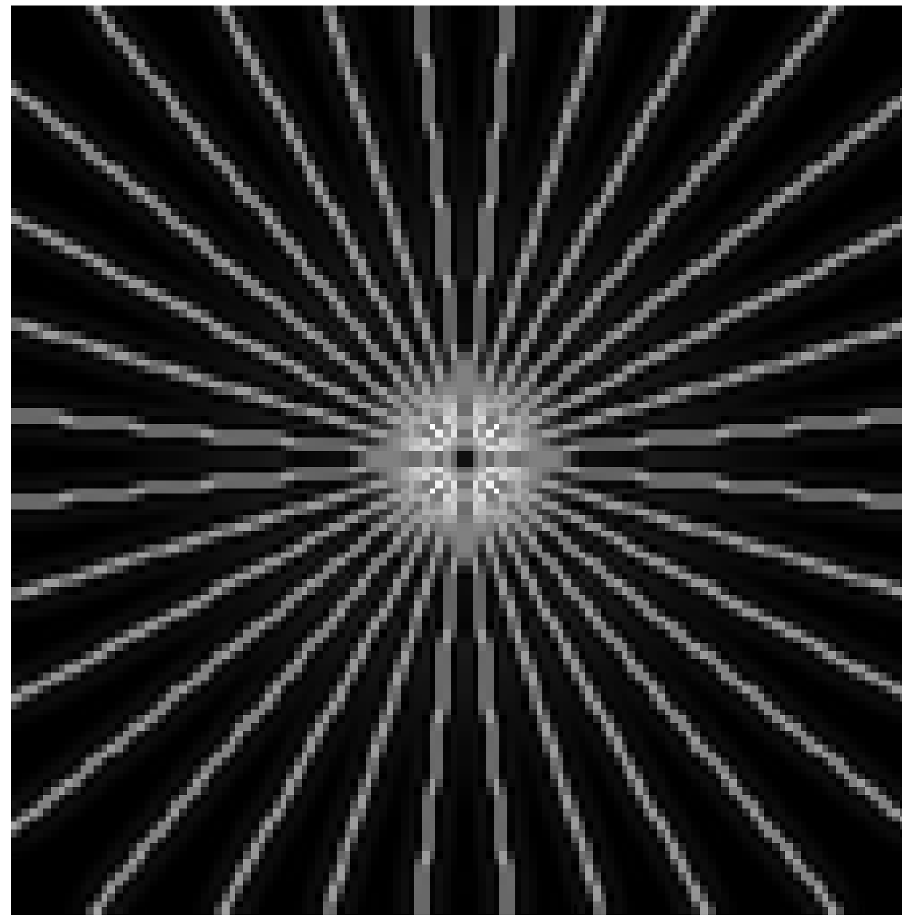


Butterworth filter

High pass filtered star



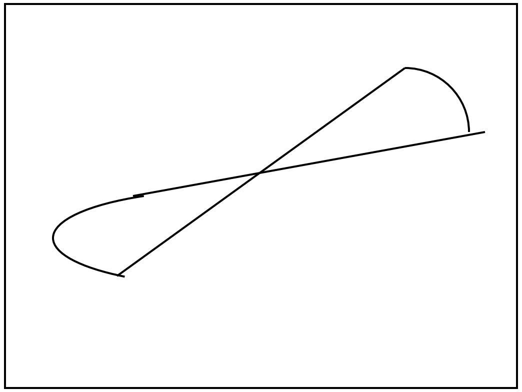
Ideal filter



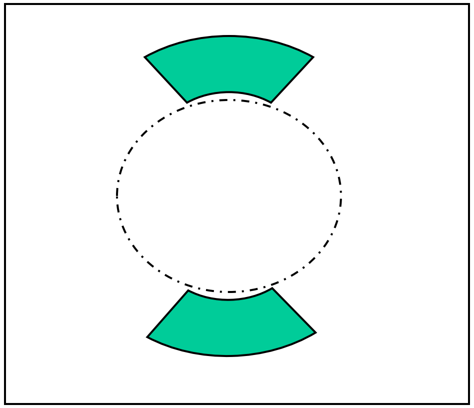
Butterworth filter

Filtering with zonal masks

- Image is a 2D signal ➔ we have additional degree of freedom in filtering
- Can design filters to select image features at specific spatial frequency and orientation
- Implementable using masks of different shapes



Fan filter



Wedge filter

Some applications

Correlation

Correlation operation

conjugate

$$f[m,n] \circ g[m,n] = y[m,n] = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f^*[i,j]\, g[m+i, n+j]$$

Popular as a template matching technique

- To find similarity between an unknown and a set of known images
- To locate instances of a given image within a larger image (ex. find an alphabet in a document image)

In the Fourier (Frequency) domain

$$\Im\{ f[m,n] \circ g[m,n] \} = \overline{F}[k,l] G[k,l]$$

Correlation can be computed with DFT <u>after zero-padding *f*</u> and *g*

Illumination correction

Image model: $f[m,n] = i[m,n]\ r[m,n]$

i – illumination (corrupted) r – reflectance (of interest)

Filtering steps

1. Convert the multiplication model to additive using a *log* transform

In the transformed space

2. Estimate the degradation $i[m,n]$
3. Subtract from given $f[m,n]$

4. Find the inverse of the *log* transform to recover the corrected image

Homomorphic filtering

Filtering steps

1. Do a Log transform of *f*: $\ln f[m,n] = \ln i[m,n] + \ln r[m,n]$

   I.e. $x[m,n] = a[m,n] + b[m,n]$

2. Compute the Fourier Transform $X[u,v] = A[u,v] + B[u,v]$

   low pass high pass

3. ***Lowpass filter*** *X* to extract the illumination component

   $LPF\{X\} \sim A \Rightarrow IDFT\{A\} = a$

4. Perform illumination correction: $y = x - a$

5. Perform inverse Log transform of *y* to obtain the desired illumination corrected image: $e^{(x-a)}$

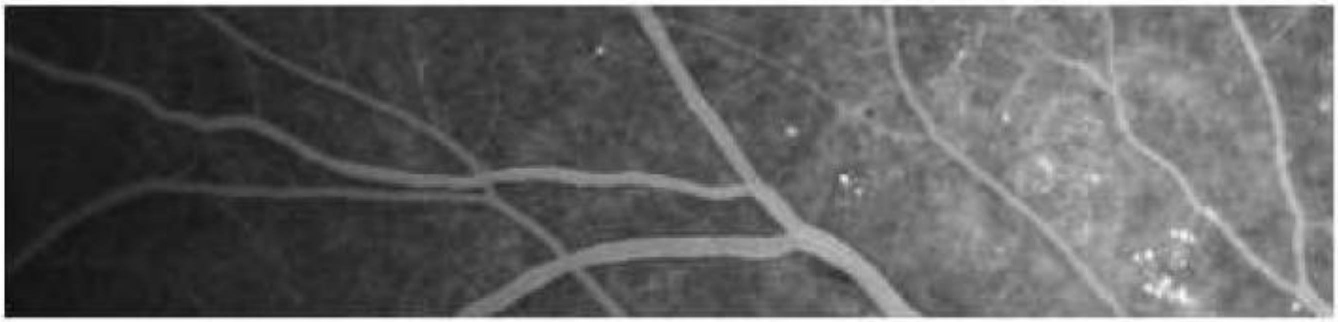
Homomorphic filtering - effects

- Corrects illumination but reduces the dynamic range
  - Lowers the brightness level of entire image

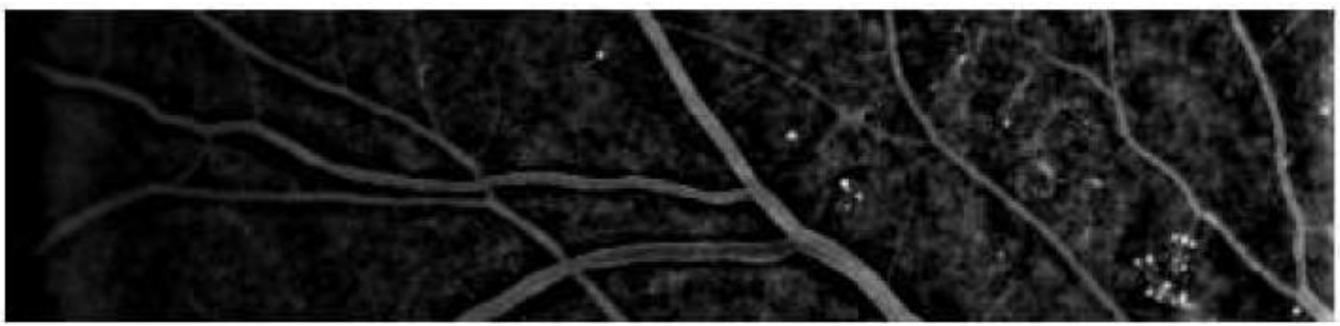
- Variable results as *f* is only estimated

Homomorphic filtering – another example

Original



Filtered



Decomposition of an image

So far we looked at

Discrete Fourier Transform

- decomposes images using a fixed basis
  - complex exponential

- What if we use a non-fixed basis?

Examples:

1. The **SVD** decomposition
   - which uses an image-adaptive basis
     - the basis is derived from the given image
2. The **Wavelet Transform**
   - which uses a non-image adaptive, non-fixed basis

SVD - Diagonalising a given image

Claim: Given any image $f(m,n)$ we can diagonalise it (make it sparse)

Let $g = ff^T$.

The eigenvectors $\boldsymbol{u}_i$ of g are found from: $g\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$

The eigenvectors $\boldsymbol{v}_i$ of g^T are found from: $g^T\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$

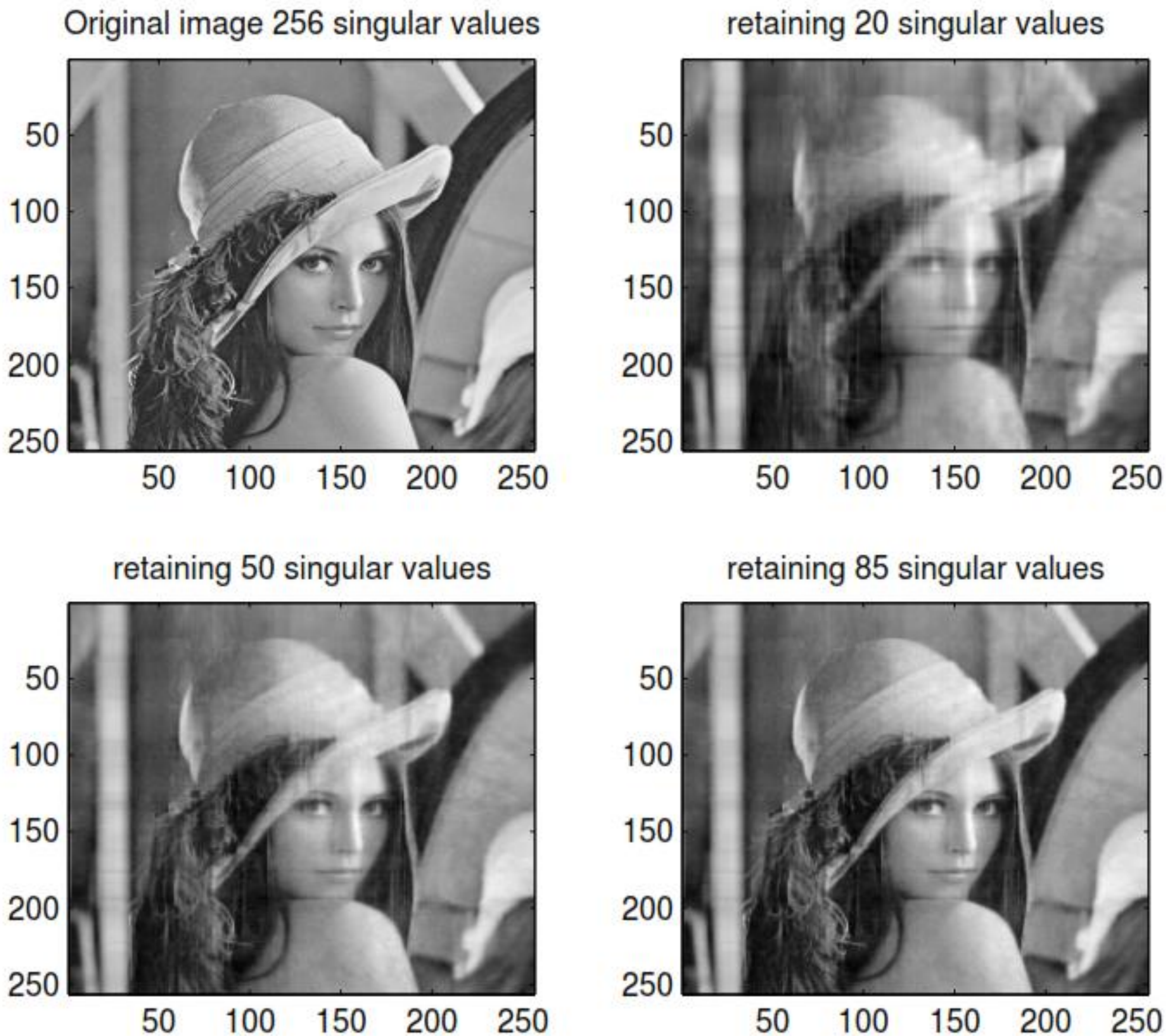
Let $U = \{\boldsymbol{u}_i\}$ and $V = \{\boldsymbol{v}_i\}$

Then, the **Singular Value Decomposition (SVD)** of f (of rank r) is

$$f = \sum_{i=1}^{r} \lambda_i^{1/2} u_i v_i^T$$ Or in matrix form $$f = U\Lambda^{1/2}V^T$$

Λ is a diagonal matrix of r non-zero eigenvalues of g

SVD based image recovery



Good solution for compression ?

From T Workalmehu 2008

SVD of images

- If Λ is constructed with decreasing λ_i then

SVD is an optimal decomposition for an image f

- f can be approximated as a linear sum of a small set (b) of basis images with least square error

- Basis images of SVD are eigenimages of f found as $\boldsymbol{u}_i \boldsymbol{v}_j^T$

Note: SVD is similar/closely related to Karhunen Loeve Transform (KLT) and Prinicipal Component Analysis (PCA)

To learn more, see Gerbrands JJ (1981) *On the relationships between SVD, KLT and PCA*, **Pattern recognition,** Vol. 14, No. 1-6. pp. 375-381.