Topic 3: Digital Sampling
Workshop Solutions

Workshop Questions

3.1 Two-dimensional Symmetry

Show that the DFT of a two-dimensional real function has the symmetry properties of

$$\begin{aligned} F_R(k,l) &= F_R(-k,-l) \\ F_R(-k,l) &= F_R(k,-l) \\ F_I(k,l) &= -F_I(-k,-l) \\ F_I(-k,l) &= -F_I(k,-l) \end{aligned}$$

Solution

The two-dimensional DFT is given by

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \exp\left(-\imath 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)\right)$$

which can be written as:

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \exp\left(-\imath 2\pi \frac{ki}{N}\right) \exp\left(-\imath 2\pi \frac{lj}{N}\right)$$

which can then be expanded in terms of cos() and sin() to give

$$F(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \left(\cos\left(2\pi\frac{ki}{N}\right) - \imath \sin\left(2\pi\frac{ki}{N}\right)\right)\left(\cos\left(2\pi\frac{lj}{N}\right) - \imath \sin\left(2\pi\frac{lj}{N}\right)\right)$$

Now if $f(i,j)$ is **real** we can collect real and imaginary parts together to give

$$F(k,l) = F_R(k,l) + \imath F_I(k,l)$$

where

$$F_R(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \left(\cos\left(2\pi\frac{ki}{N}\right)\cos\left(2\pi\frac{lj}{N}\right) - \sin\left(2\pi\frac{ki}{N}\right)\sin\left(2\pi\frac{lj}{N}\right)\right)$$

and

$$F_I(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \left(\cos\left(2\pi\frac{ki}{N}\right)\sin\left(2\pi\frac{lj}{N}\right) + \cos\left(2\pi\frac{lj}{N}\right)\sin\left(2\pi\frac{ki}{N}\right)\right)$$

The symmetry relation is given by the cos() and sin() functions. Substitute,

$$a = 2\pi\frac{ki}{N} \quad \& \quad b = 2\pi\frac{lj}{N}$$

so the symmetry for the *real* part is given by:

$$\cos(a)\cos(b) - \sin(a)\sin(b)$$

so that, noting that $\cos(a) = cos(-a)$ and $\sin(a) = -\sin(-a)$,

$$F_R(a,b) = F_R(-a,-b) \quad \& \quad F_R(-a,b) = F_R(a,-b)$$

and the *imaginary* part by:

$$\cos(a)\sin(b) + \sin(a)\cos(b)$$

so that:

$$F_I(a,b) = -F_I(-a,-b) \quad \& \quad F_I(-a,b) = -F_I(a,-b)$$

as expected.

3.2 Symmetry Pairing

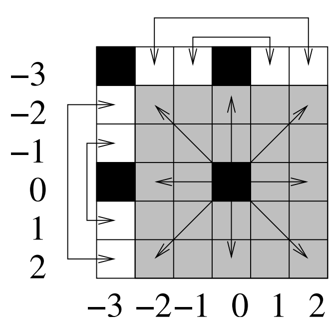
Verify for a 6×6 image that the DFT of a two-dimensional real function has:

$$\frac{N^2}{2} + 2 \qquad \text{Independent real values}$$
$$\frac{N^2}{2} - 2 \qquad \text{Independent imaginary values}$$

Fourier filters involve multiplying the DFT by a filtering function $H(i, j)$. Many of these filters are real only. Suggest a scheme for packing the real and imaginary parts of a DFT into a square array that makes multiplication with such a filter simple.

Solution

First renumber the Fourier array so that the $(k : l = -3, \rightarrow, 2)$ which can be done due to the cyclic properties of the DFT. There are now **2** arrays of 36 element, one for the *real* parts and one form the *imaginary* parts as shown below.



Real Part: is symmetric. Consider three regions of the output place, shaded *grey, black & white* above.

1. *grey*: There are 24 points where the symmetric symmetry is simple: eg: $F_R(1,2) = F_R(-1,-2)$. So in this regions there are **12** unique values with the other *12* given by symmetry.

2. *black*: There are 4 points with no symmetric pair, actually due to the cylcic properties they are their *own* symmetric pair. This gives **4** uniques values.

3. *white*: There are 8 points in 4 symmetric pairs but they are wrapped round due to cyclic property. For example $F_R(-3,2) = F_R(3,-2)$ but $F_R(3,-2)$ is cyclic with period 6, so that $F_R(3,-2) = F_R(-3,-2)$. This region gives **4** uniques values with the other *4* given by symmetry.

This gives total of:

Area	Number
grey	12
black	4
white	4
Total	**20**

Imaginary Part: is anit-symmetric. Consider same three regions,

1. *grey*: As for *real* there are 24 points where symmetry is simple: eg: $F_I(1,2) = -F_I(-1,-2)$. So in this regions there are **12** unique values with the other *12* given by symmetry.

2. *black*: There are 4 points where all terms in the expression for the *imaginary* componets contain $\sin(\pm n\pi) = 0$. So all these 4 points are always zero and to not depend on $f(i,j)$ This area contributes **0** values.

3. *white*: There are 8 points in 4 symmetric pairs but they are wrapped round due to cyclic property. This region gives **4** uniques values with the other *4* given by symmetry.

This gives total of:

Area	Number
grey	12
black	0
white	4
Total	**16**

So as noted we have

$$\begin{aligned} \frac{N^2}{2} + 2 &= 20 \quad \text{Real Values} \\ \frac{N^2}{2} - 2 &= 16 \quad \text{Imaginary Values} \end{aligned}$$

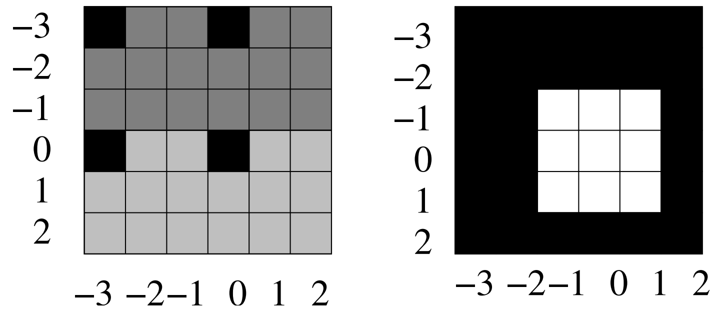
There are 36 values that can be "packed" into a single 6×6 float array. This was shown for $N = 4$ in lectures, $N = 6$ here, and can (*easily*) be extended to any **even** N.

If we want to apply a *real* filter $H(k,l)$ then we need to multiply both the *real* and *imaginary* parts by the same filter to give:

$$G_R(k,l) = F_R(k,l)\,H(k,l) \quad \& \quad G_I(k,l) = F_I(k,l)\,H(k,l)$$

we can then inverse transform to give the filtered image. (For details of filter types see lecture 8). In order to preserve the symmetry of $G(k,l)$ the filter $H(k,l)$ **must** be symmetric. If we then pack the Fourier image data $F(k,l)$ as shown on left, with the

1. *Real Symmetric* part in the *Dark Grey* region.
2. *Imaginary Anti-symmetric* part in the *Light Grey* region.
3. *Additional 4 Real Values* in the *Black* regions.



Then if we apply a symmetric filter, as shown on the right, then the apperant multiplication reduced to a $N \times N$ real multiplication with the "top" half of the filter multiplying the *real* section and the "bottom" half the *imaginary* section.

3.3 Shifting The Centre

Show that if your two dimensional DFT code locates the $(0,0)$ term in the top/left of the array, then this can be shifted to the centre of the array by pre-multiplying the by a ± 1 checker-board.

Solution

Want to shift the *top-left* pixel to the centre of $N \times N$ array which occurs at $(N/2, N/2)$. Mathematically a *shift* can be implemented as a convolution with a shifted δ-function, so we want to form:

$$F(k,l) \odot \delta\left(k - \frac{N}{2}, l - \frac{N}{2}\right)$$

(Note that we want cyclic wrap-round so that sections of $F(i, j)$ with $i, j > N/2$ will wrap-round to the $i, j < N/2$ sections. This will occur automatically due to the cyclic property of the DFT.)

The convolution in Fourier Space is equivalent to a multiplication in Real space of

$$f(i, j)\mathcal{F}\left\{\delta\left(k - \frac{N}{2}, l - \frac{N}{2}\right)\right\}$$

We have to take the DFT, so

$$\mathcal{F}\{\delta(k-a)\} = \frac{1}{N}\sum_{i=0}^{N-1} \delta(k-a)\exp\left(-\imath 2\pi \frac{ik}{N}\right) = \frac{1}{N}\exp\left(-\imath 2\pi \frac{ai}{N}\right)$$

so in two dimensions we have that:

$$\mathcal{F}\left\{\delta\left(k - \frac{N}{2}, l - \frac{N}{2}\right)\right\} = \frac{1}{N^2}\exp\left(\imath\pi(i+j)\right)$$

Ignoring the $1/N^2$ term, we note that

$$\begin{aligned} \exp(\imath\pi(i+j)) &= 1 \quad \text{when } (i+j) \text{ Even} \\ &= -1 \quad \text{when } (i+j) \text{ Odd} \end{aligned}$$

so for N even we have to multiply the function $f(i,j)$ by

1	−1	1	⋯	−1
−1	1	−1	⋯	1
⋮	⋮	⋮	⋱	⋮
1	−1	1	⋯	−1
−1	1	−1	⋯	1

3.4 Speed of the FFT

On a particular computer system the FFT of a 128×128 image takes 0.11 seconds, estimate how long this system would take to calculate the FFT of a 1024×1024 image.

Solution

For a one-dimensional FFT the computational cost is proportional to $N\log_2(N)$. The two-dimensional DFT can be formulated at $2N$ one-dimensional FFT (N along the rows, plus N down the columns). The computational cost for the FFT of an $N \times N$ array is therefore

$$\propto N^2 \log_2(N)$$

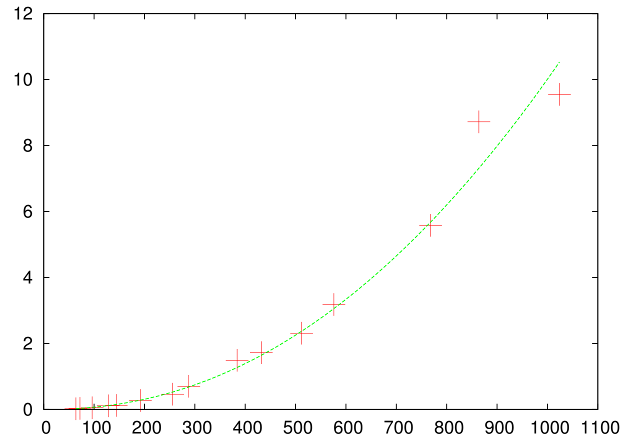
If it takes 0.11 secs for $N = 128$ then for $N = 1024$ is will take

$$\frac{1024^2 \log_2(1024)}{128^2 \log_2(128)} \times 0.11 = 10.05\text{secs}$$

Real experimental values for $N = 2^n$ are:

Size	Time in secs.
64	0.02
128	0.11
256	0.46
512	2.31
1024	9.55

which is very close to the predicted timings. Experimental timings for various N consisting of powers or 2 & 3 from 64 → 1024 are shown in the graph below.



The best fit of $N^2 \log_2(N)$ is also plotted. The $N = 2^n$ points are all *below* the curve while the values of N containing 3(s) are above the line. This shows that this algorithm is optimally efficient for powers-of-two.

Timings for larger N showed very significant deviation for expected $N^2 \log_2(N)$ relation due to size of physical memory, for example a 2048×2048 DFT took 20 minutes.

3.5 CCD Sensors

A CCD sensor is a two-dimensional array of detectors that can be used to sample an image. A typical TV quality CCD camera will have 586×768 sensors on a 15 by 20 mm area with a $3:4$. Calculate the size of the sensors and the maximum spatial frequency in the detected image.

You wish to use this CCD camera to image pages of text for a character recognition system that is able to easily resolved 8pt (1pt is 1/72nd of an inch) letters. What magnification is required and how large a page of text can be images at once.

Hint: To easily resolve a letter you must be able to resolve line approximately 5 times closer together than the minimum separation of lines in the letter.

Solution

Video images have a 4:3 aspect ratio due to the shape of the normal TV screen. The sensor sizes are simply

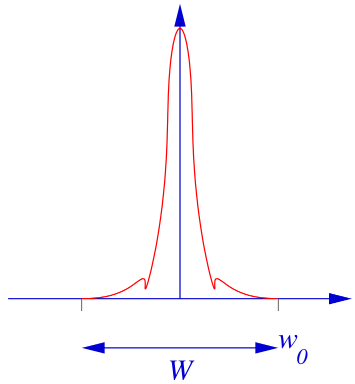
$$\frac{15}{586} = 25.6\mu\text{m} \qquad \frac{20}{768} = 26.0\mu\text{m}$$

so, for all practical purposes, they are square.

Shannon sampling rate is that the signal is optimally sampled with sampling interval Δx if

$$\Delta x = \frac{1}{W}$$

where W is the “bandwidth” of the signal.



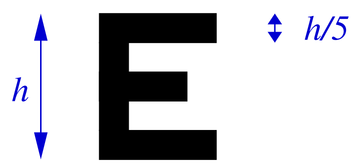
The *maximum* spatial frequency is thus *half* the bandwidth, so that

$$w_0 = \frac{1}{2}W = \frac{1}{2\Delta x}$$

so the maximum spatial frequencies are, u_0, v_0,

$$u_0 = 19.2\text{lines}/\text{mm} \quad \& \quad v_0 = 19.5\text{lines}/\text{mm}$$

Letter in an 8pt font are approximately 2.8 mm high. This is typically the smallest font that appears in printed documents.



The typical lines in the letter are about 1/5th of the letter's height, but to "easily" resolve the letter we must be able to resolve lines about 5 times closer together than that so we must resolve lines with a separation of

$$\frac{2.8}{25} = 112\mu\text{m}$$

which is equivalent to spatial frequency of:

$$8.93\text{lines}/\text{mm}$$

The CCD sensor has a maximum spatial frequency resolution of approximately 19 lines/mm, so the magnification between object and image plane is given by

$$\frac{8.93}{19.2} = 0.46$$

in other words the imaged area of the page is approximately $\times 2.15$ the size of the sensor, so the size of the imaged "page" is approximately,

$$43 \times 32\,\text{mm}$$

which is only a small section of a page. In a typical book an area of this size contains about 45 words.

This calculation shows that optical character recognition requires a vast amount of data to be processed. Real character recognition systems do not use video cameras by a linear sensor array the width of the paper.