Discrete Fourier Transform

Nuno Vasconcelos
UCSD

Fourier Transforms

- we started by considering the Discrete-Space Fourier Transform (DSFT)
- the DSFT is the 2D extension of the Discrete-Time Fourier Transform

$$X(\omega_1,\omega_2)=\sum_{n_1}\sum_{n_2}x[n_1,n_2]e^{-j\omega_1 n_1}e^{-j\omega_2 n_2}$$

$$x[n_1,n_2]=\frac{1}{(2\pi)^2}\iint X(\omega_1,\omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2}d\omega_1 d\omega_2$$

- note that this is a continuous function of frequency
  - inconvenient to evaluate numerically in DSP hardware
  - we need a discrete version
  - this is the 2D Discrete Fourier Transform (2D-DFT)

2D-DFT

- the 2D-DFT is obtained by sampling the DSFT at regular frequency intervals

$$X[k_1,k_2] = X(\omega_1,\omega_2)\Big|_{\omega_1=\frac{2\pi}{N_1}k_1,\,\omega_2=\frac{2\pi}{N_2}k_2}$$

- this turns out to make the 2D-DFT harder to work with than the DSFT
  - because we are sampling in frequency we have aliasing in space
  - this means that, even though the sequence $x[n_1,n_2]$ is finite, we are effectively working with a periodic sequence
  - the DFT therefore inherits all the properties of the frequency representations of periodic sequences
- it is better understood by first considering the 2D Discrete Fourier Series (2D-DFS)

2D-DFS

- it is the natural representation for a periodic sequence
- a sequence $\underline{x}[n_1,n_2]$ is periodic of period $N_1 x N_2$ if

$$\begin{aligned}\underline{x}[n_1,n_2] &= \underline{x}[n_1+N_1,n_2] \\ &= \underline{x}[n_1,n_2+N_2], \quad \forall n_1,n_2\end{aligned}$$

- note that

$$\underline{X}(r_1,r_2) = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} x[n_1,n_2] r_1^{-jn_1} r_2^{-jn_2}$$

- makes no sense for a periodic signal
  - the sum will be infinite for any pair r_1,r_2
  - neither the 2D DSFT or the Z-transform will work here

2D-DFS

- the 2D-DFS solves this problem

$$\underline{X}(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\underline{x}[n_1,n_2]e^{-j\frac{2\pi}{N_1}k_1n_1}e^{-j\frac{2\pi}{N_2}k_2n_2}$$

$$\underline{x}[n_1,n_2]=\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}\underline{X}[k_1,k_2]e^{j\frac{2\pi}{N_1}k_1n_1}e^{j\frac{2\pi}{N_2}k_2n_2}$$

- note that $\underline{X}[k_1,k_2]$ is also periodic outside

$$0\le k_1\le N_1-1,\ \ 0\le k_2\le N_2-1$$

- like the DSFT,
  - properties of the 2D-DFS are identical to those of the 1D-DFS
  - with the straightforward extension of separability

Periodic convolution

- like the Fourier transform,
  - the inverse transform of multiplication is convolution

$$\underline{x}[n_1, n_2] * \underline{x}[n_1, n_2] \overset{DFS}{\leftrightarrow} \underline{X}(k_1, k_2) \times \underline{Y}(k_1, k_2)$$

  - however, we have to be careful about how we define convolution
  - since the sequences have no end, the standard definition

$$y[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h[n_1 - k_1, n_2 - k_2]$$

    makes no sense

  - e.g. if x and h are both positive sequences, this will always be infinite

Periodic convolution

- to deal with this, we introduce the idea of periodic convolution
- instead of the regular definition

$$x * y = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, h[n_1 - k_1, n_2 - k_2]$$

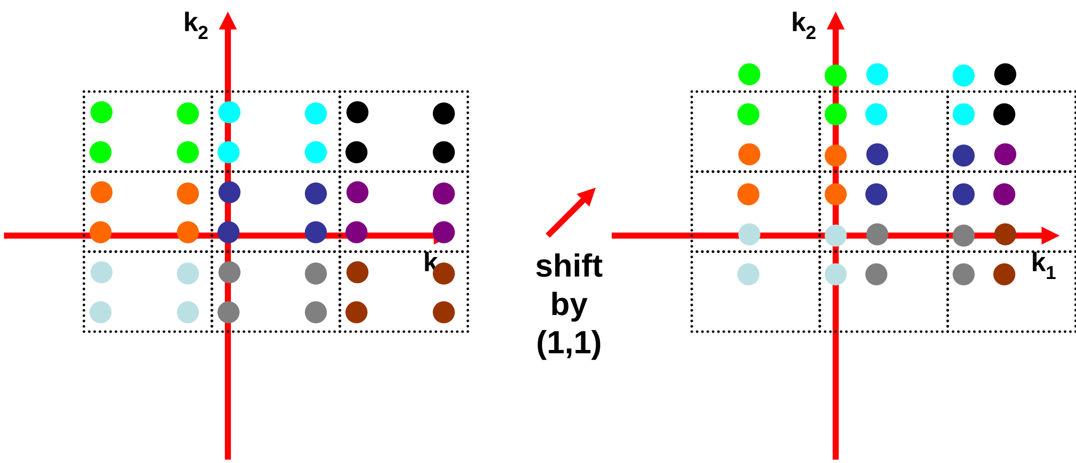
- which, from now on, we refer to as linear convolution
- periodic convolution only considers one period of our sequences

$$x \circ h = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} x[k_1, k_2]\, h[n_1 - k_1, n_2 - k_2]$$

- the only difference is in the summation limits

Periodic convolution

- note that the sequence which results from the convolution is also periodic
- it is important to remember the following
  - we work with a single period (the fundamental period) to make things manageable
  - but remember that we have periodic sequences
  - it is like if we were peeking through a window
  - if we shift, or flip the sequence we need to remember that
  - the sequence does not simply move out of the window, but the next period walks in!!!
  - note, that this can make the fundamental period change considerably



Discrete Fourier Transform

- the DFT is defined as

$$X[k_1,k_2] = X(\omega_1,\omega_2)\Big|_{\omega_1=\frac{2\pi}{N_1}k_1,\,\omega_2=\frac{2\pi}{N_2}k_2}$$

(here $X(\omega_1,\omega_2)$ is the DSFT) which can be written as

$$X[k_1,k_2] = \begin{cases} \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x[n_1,n_2]\, e^{-j\frac{2\pi}{N_1}k_1n_1} e^{-j\frac{2\pi}{N_2}k_2n_2}, & \begin{matrix} 0 \le k_1 < N_1 \\ 0 \le k_2 < N_2 \end{matrix} \\ 0 & otherwise \end{cases}$$

$$x[n_1,n_2] = \begin{cases} \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X[k_1,k_2]\, e^{j\frac{2\pi}{N_1}k_1n_1} e^{j\frac{2\pi}{N_2}k_2n_2} & \begin{matrix} 0 \le n_1 < N_1 \\ 0 \le n_2 < N_2 \end{matrix} \\ 0 & otherwise \end{cases}$$

Discrete Fourier Transform

- comparing this

$$X[k_1,k_2]=\begin{cases}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]e^{-j\frac{2\pi}{N_1}k_1n_1}e^{-j\frac{2\pi}{N_2}k_2n_2}, & \begin{matrix}0\le k_1<N_1\\0\le k_2<N_2\end{matrix}\\ 0 & otherwise\end{cases}$$

$$x[n_1,n_2]=\begin{cases}\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}X[k_1,k_2]e^{j\frac{2\pi}{N_1}k_1n_1}e^{j\frac{2\pi}{N_2}k_2n_2} & \begin{matrix}0\le n_1<N_1\\0\le n_2<N_2\end{matrix}\\ 0 & otherwise\end{cases}$$

with the DFS

$$\underline{X}[k_1,k_2]=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\underline{x}[n_1,n_2]e^{-j\frac{2\pi}{N_1}k_1n_1}e^{-j\frac{2\pi}{N_2}k_2n_2}$$

$$\underline{x}[n_1,n_2]=\frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}\underline{X}[k_1,k_2]e^{j\frac{2\pi}{N_1}k_1n_1}e^{j\frac{2\pi}{N_2}k_2n_2}$$

Discrete Fourier Transform

- we see that inside the boxes

$$\begin{matrix} 0 \le k_1 < N_1 \\ 0 \le k_2 < N_2 \end{matrix} \qquad \qquad \begin{matrix} 0 \le n_1 < N_1 \\ 0 \le n_2 < N_2 \end{matrix}$$

  the two transforms are exactly the same

- if we define the indicator function of the box

$$R_{N_1 \times N_2}[n_1, n_2] = \begin{cases} 1, & \begin{matrix} 0 \le n_1 < N_1 \\ 0 \le n_2 < N_2 \end{matrix} \\ 0 & otherwise \end{cases}$$

- we can write

$$x[n_1, n_2] = \underline{x}[n_1, n_2] R_{N_1 \times N_2}[n_1, n_2] \qquad X[k_1, k_2] = \underline{X}[k_1, k_2] R_{N_1 \times N_2}[k_1, k_2]$$

Discrete Fourier Transform

- note from

$$x[n_1, n_2] = \underline{x}[n_1, n_2] R_{N_1 \times N_2}[n_1, n_2] \qquad X[k_1, k_2] = \underline{X}[k_1, k_2] R_{N_1 \times N_2}[k_1, k_2]$$

  that working in the DFT domain is equivalent to
  - working in the DFS domain
  - extracting the fundamental period at the end
- we can summarize this as

$$x[n_1, n_2] \underset{truncate}{\overset{periodicize}{\rightleftarrows}} \underline{x}[n_1, n_2] \overset{DFS}{\leftrightarrow} \underline{X}[k_1, k_2] \underset{periodicize}{\overset{truncate}{\rightleftarrows}} X[k_1, k_2]$$

- in this way, I can work with the DFT without having to worry about aliasing

Discrete Fourier Transform

$$x[n_1,n_2] \underset{truncate}{\overset{periodicize}{\rightleftarrows}} \underline{x}[n_1,n_2] \overset{DFS}{\leftrightarrow} \underline{X}[k_1,k_2] \underset{periodicize}{\overset{truncate}{\rightleftarrows}} X[k_1,k_2]$$

- this trick can be used to derive all the DFT properties
- e.g. what is the inverse transform of a phase shift?
  - let's follow the steps

$$Y[k_1,k_2] = X[k_1,k_2] e^{-j\frac{2\pi}{N_1}k_1 m_1} e^{-j\frac{2\pi}{N_2}k_2 m_2}$$

  - 1) periodicize: this causes the same phase shift in the DFS

$$\underline{Y}[k_1,k_2] = \underline{X}[k_1,k_2] e^{-j\frac{2\pi}{N_1}k_1 m_1} e^{-j\frac{2\pi}{N_2}k_2 m_2}$$

Discrete Fourier Transform

$$x[n_1,n_2] \;\underset{\text{truncate}}{\overset{\text{periodicize}}{\overset{\longrightarrow}{\longleftarrow}}}\; \underline{x}[n_1,n_2] \;\overset{DFS}{\longleftrightarrow}\; \underline{X}[k_1,k_2] \;\underset{\text{periodicize}}{\overset{\text{truncate}}{\overset{\longrightarrow}{\longleftarrow}}}\; X[k_1,k_2]$$

– 2) compute the inverse DFS: it follows from the properties of the DFS (page 142 on Lim) that we get a shift in space

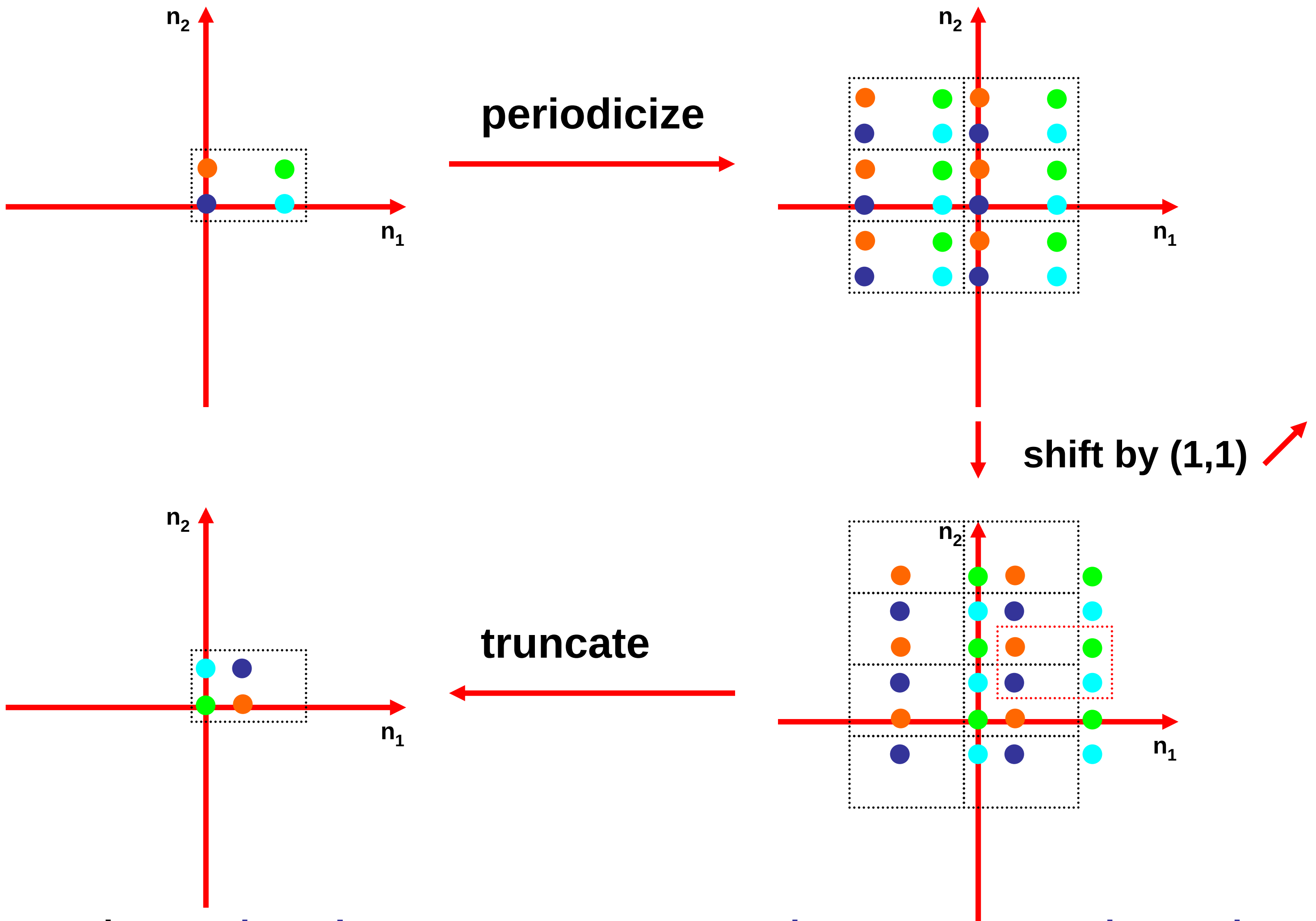
$$\underline{y}[n_1,n_2] = \underline{x}[n_1 - m_1, n_2 - m_2]$$

– 3) truncate: the inverse DFT is equal to one period of the shifted periodic extension of the sequence

$$y[n_1,n_2] = \underline{x}[n_1 - m_1, n_2 - m_2] R_{N_1 \times N_2}[n_1,n_2]$$

– in summary, the new sequence is obtained by making the original periodic, shifting, and taking the fundamental period

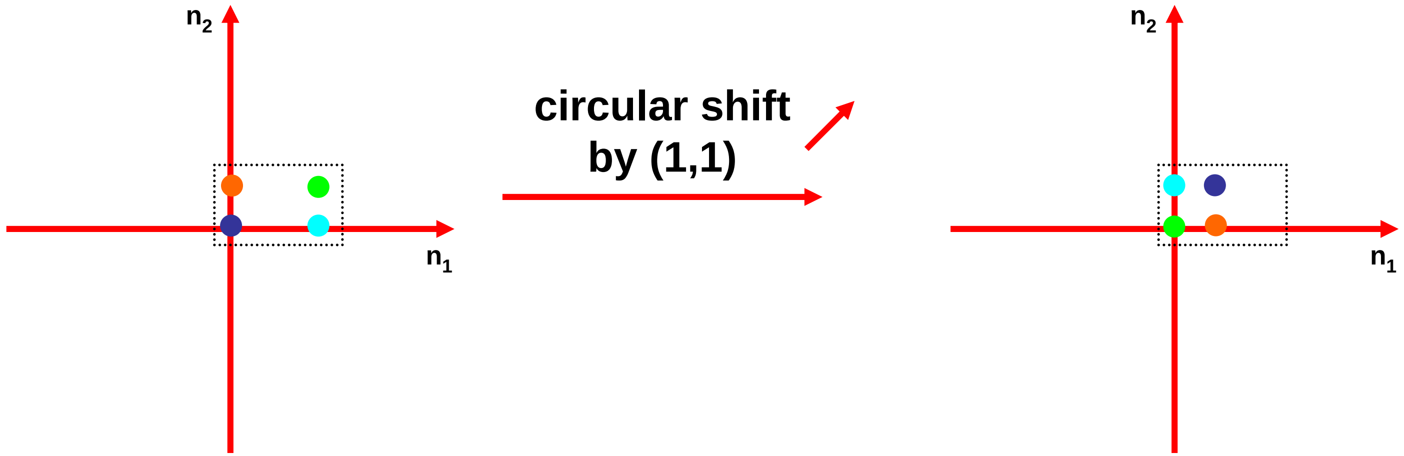
Example



- note that what leaves on one end, enters on the other

Example

- for this reason it is called a circular shift



- note that this is way more complicated than in 1D
- to get it right we really have to think in terms of the periodic extension of the sequence
- it shows up in most properties of the DFT,
- e.g. what is the inverse DFT of the product of two DFTs?

Discrete Fourier Transform

$$x[n_1,n_2] \underset{\textit{truncate}}{\overset{\textit{periodicize}}{\rightleftarrows}} \underline{x}[n_1,n_2] \overset{DFS}{\leftrightarrow} \underline{X}[k_1,k_2] \underset{\textit{periodicize}}{\overset{\textit{truncate}}{\rightleftarrows}} X[k_1,k_2]$$

- we use our trick again

$$Y[k_1,k_2] = X[k_1,k_2]H[k_1,k_2]$$

  - 1) periodicize:

$$\underline{Y}[k_1,k_2] = \underline{X}[k_1,k_2]\underline{H}[k_1,k_2]$$

  - 2) compute the inverse DFS: this is just the periodic convolution

$$\underline{y}[n_1,n_2] = \underline{x}[n_1,n_2] \circ \underline{y}[n_1,n_2]$$

Discrete Fourier Transform

$$x[n_1,n_2] \underset{\text{truncate}}{\overset{\text{periodicize}}{\rightleftarrows}} \underline{x}[n_1,n_2] \overset{DFS}{\leftrightarrow} \underline{X}[k_1,k_2] \underset{\text{periodicize}}{\overset{\text{truncate}}{\rightleftarrows}} X[k_1,k_2]$$

– 3) truncate: the inverse DFT is equal to one period of the periodic convolution of the sequences

$$y[n_1,n_2] = \left(\underline{x}[n_1,n_2] \circ \underline{h}[n_1,n_2]\right) R_{N_1 \times N_2}[n_1,n_2]$$

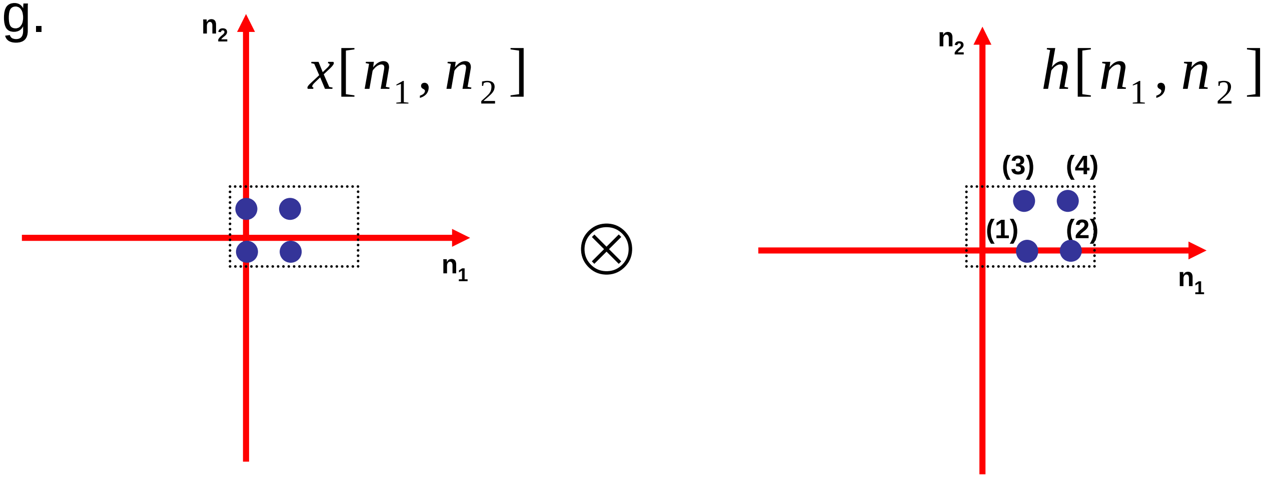
– in summary, the new sequence is obtained by making the original sequences periodic, computing the periodic convolution, and taking the fundamental period

– this is the circular convolution of $x[n_1,n_2]$ and $h[n_1,n_2]$

$$x[n_1,n_2] \otimes h[n_1,n_2] = \left(\underline{x}[n_1,n_2] \circ \underline{h}[n_1,n_2]\right) R_{N_1 \times N_2}[n_1,n_2]$$

Discrete Fourier Transform

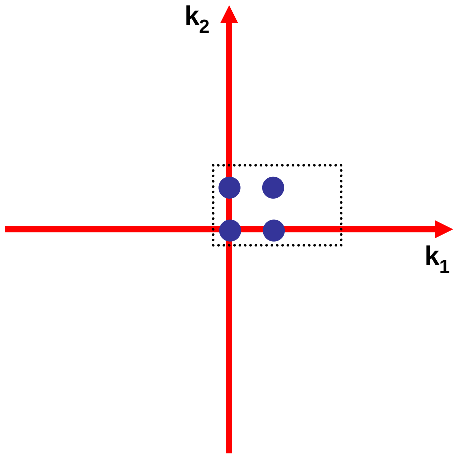
- we therefore have the property that
  - the product of two DFTs is the
  - DFT of the circular convolution of the two sequences
- note that circular convolution = one period of periodic convolution
- hence, there is really not much that is new
  - periodicize the sequences, and apply what we learned for the convolution of DFSs
  - e.g.



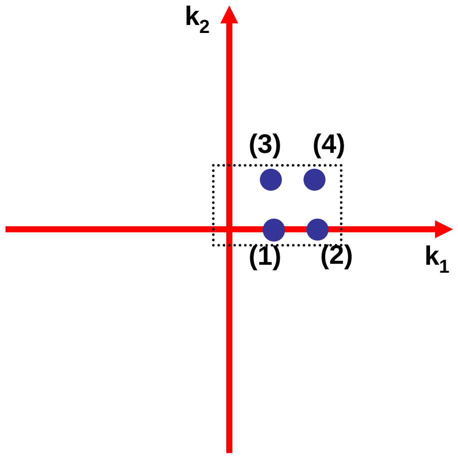
Circular convolution

- **step 1):** express sequences in terms of *(k_1, k_2)*,

$$x[n_1,n_2]\otimes h[n_1,n_2]=\left(\underline{x}[n_1,n_2]\circ \underline{h}[n_1,n_2]\right)R_{N_1\times N_2}[n_1,n_2]$$



$x[k_1,k_2]$



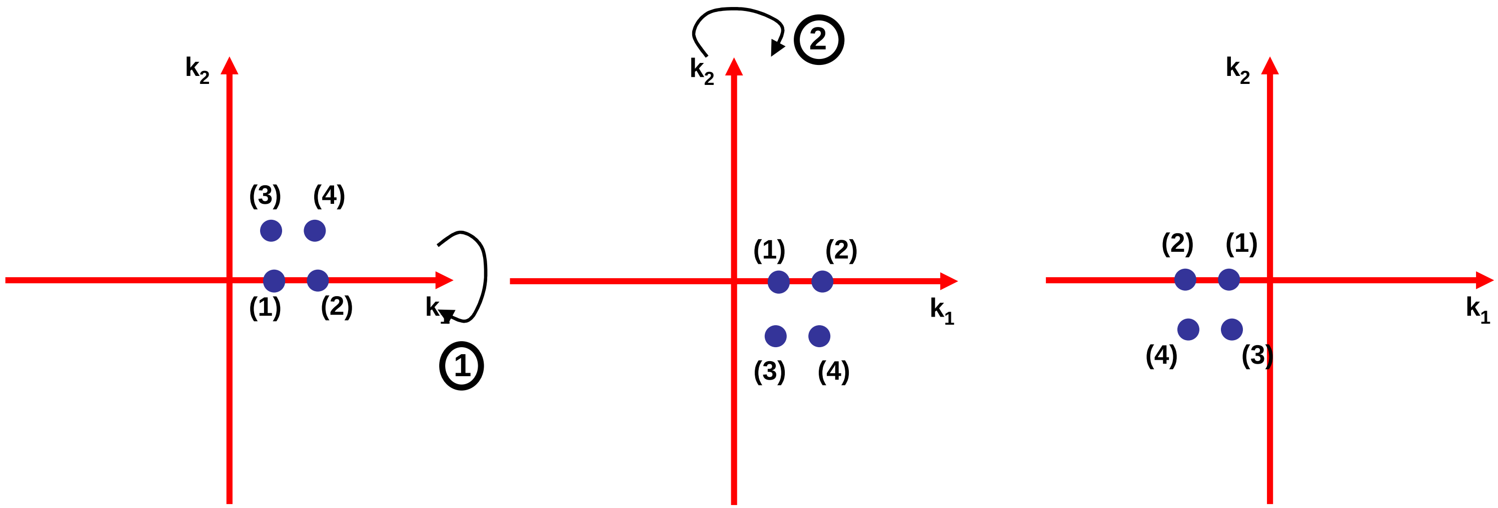
$h[k_1,k_2]$

we next proceed exactly as for periodic convolution

Circular convolution

- **step 2):** invert *h(k₁, k₂)*

$$\underline{x} \circ \underline{h} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \underline{x}[k_1, k_2]\, \underline{h}[n_1 - k_1, n_2 - k_2]$$

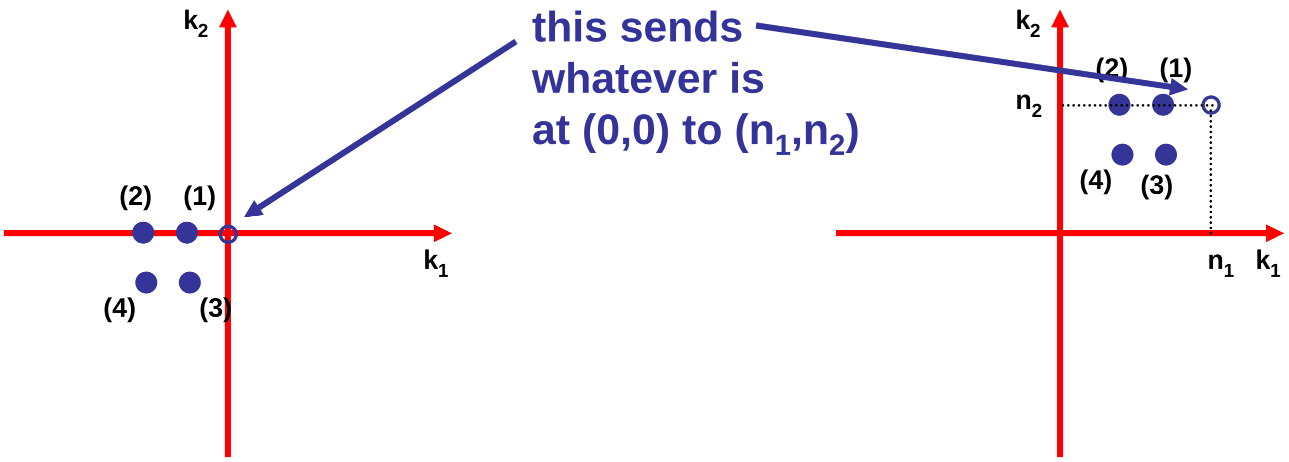


$h[k_1, k_2]$ $h[k_1, -k_2]$ $g[k_1, k_2] = h[-k_1, -k_2]$

Circular convolution

- **step 3):** shift *g(k₁, k₂)* by *(n₁, n₂)*

$$\underline{x} \circ \underline{h} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \underline{x}[k_1, k_2]\, \underline{h}[n_1 - k_1, n_2 - k_2]$$

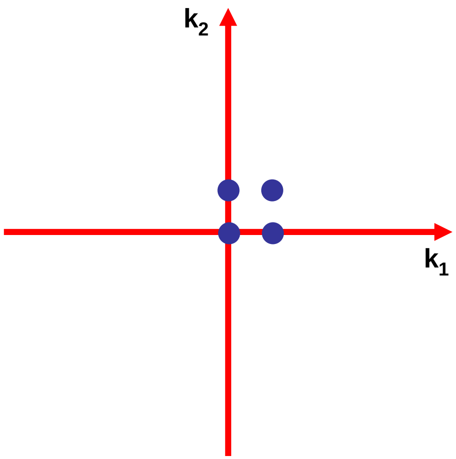


$$g[k_1, k_2] = h[-k_1, -k_2]$$

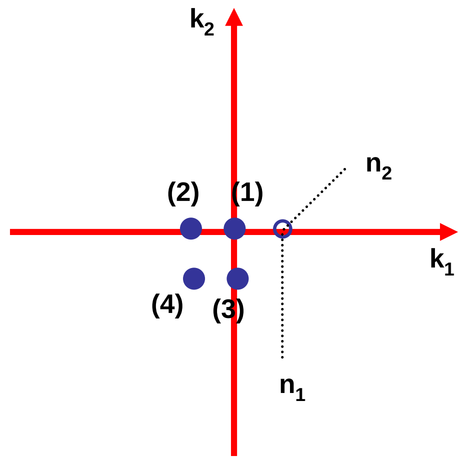
$$g[k_1 - n_1, k_2 - n_2] = h[n_1 - k_1, n_2 - k_2]$$

Circular convolution

- e.g. for $(n_1,n_2) = (1,0)$



$$x[k_1, k_2]$$



$$h[n_1 - k_1, n_2 - k_2]$$

but here we

- recall that we are working with periodic sequences
- use periodicity to fill values missing in the flipped sequence

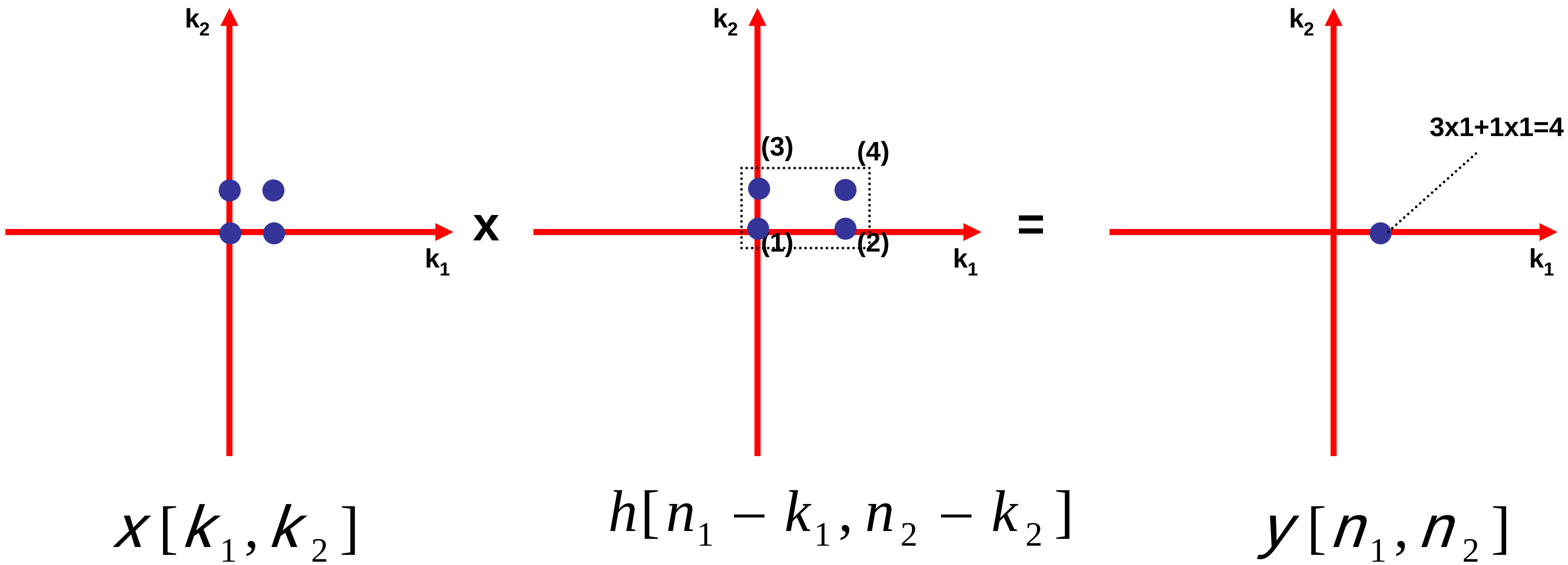
$$\underline{x} \circ \underline{h} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \underline{x}[k_1, k_2]\, \underline{h}[n_1 - k_1, n_2 - k_2]$$

Circular convolution

- **step 4):** we can finally point-wise multiply the two signals and sum

$$\underline{x} \circ \underline{h} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \underline{x}[k_1, k_2]\, \underline{h}[n_1 - k_1, n_2 - k_2]$$

  – e.g. for $(n_1,n_2) = (1,0)$



Circular convolution

- finally, we extract the fundamental period

$$x[n_1, n_2] \otimes h[n_1, n_2] = \left(\underline{x}[n_1, n_2] \circ \underline{h}[n_1, n_2]\right) R_{N_1 \times N_2}[n_1, n_2]$$

- note that the sequence never grows beyond our original window
- this is fundamentally different from linear convolution
  - it is the reason why we need to do circular shifts
- note that, because of this, it can be very different to
  - 1) convolve two signals
  - 2) take the DFTs, multiply, and take inverse DFT
- let's see what happens on MATLAB

Circular convolution

- >> x = [1 2 3; 3 3 1; 1 5 5]; h = [0 3 4; 5 2 1; 1 3 2]; z = conv2(x,h)
- z =
- 0 3 10 17 12
- 5 21 41 23 7
- 16 29 44 53 27
- 8 39 52 24 7
- 1 8 22 25 10

- >> H = fft2(h); X = fft2(x); Y = X.*H; y = ifft2(Y)
- y =
- 49 61 62
- 54 46 63
- 69 56 44

Discrete Fourier Transform

- why do we care about the DFT?
  - 1) we need a discrete representation of the frequency spectrum if we are to implement algorithms on computers
  - the DSFT cannot be used for this because it is continuous
  - 2) there are very fast algorithms to compute the DFT
- in 1D DSP you may have mentioned the Fast Fourier Transform (FFT)
  - it is a fast algorithm to compute the DFT
  - if the sequence has N points, instead of $O(N^2)$ complexity, it has $O(N \log N)$
  - this has made a tremendous historical difference
  - FFT speedup = one or two generations of DSP hardware
- Q: is there a two dimensional FFT?

Fast Fourier Transform

- to answer this we look at the expression of the DFT

$$X[k_1,k_2]=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]e^{-j\frac{2\pi}{N_1}k_1n_1}e^{-j\frac{2\pi}{N_2}k_2n_2}$$

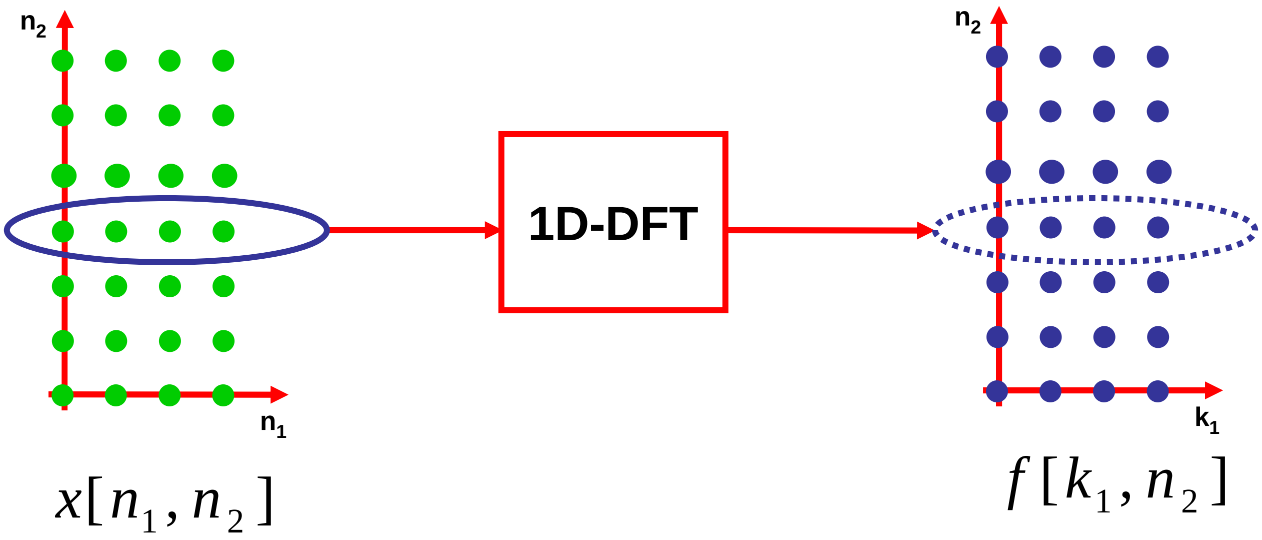
- note that this can be computed with

$$X[k_1,k_2]=\sum_{n_2=0}^{N_2-1}e^{-j\frac{2\pi}{N_2}k_2n_2}\underbrace{\left\{\sum_{n_1=0}^{N_1-1}x[n_1,n_2]e^{-j\frac{2\pi}{N_1}k_1n_1}\right\}}_{f(k_1,n_2)}$$

- given n_2, $f[k_1,n_2]$ is the 1D DFT of $x[n_1,n_2]$
  - i.e. the 1D-DFT of row n_2 of the sequence x
- we have seen something like this when we studied separability

Fast Fourier Transform

- the idea is to create an intermediate sequence $f[k_1,n_2]$
  - whose rows are the DFTs of the rows of x

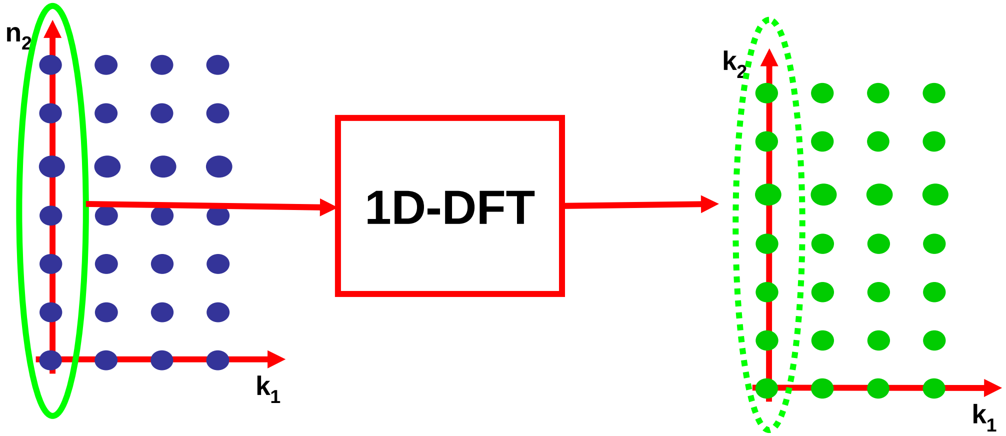


- next we realize that

$$X[k_1,k_2] = \sum_{n_2=0}^{N_2-1} e^{-j\frac{2\pi}{N_2}k_2 n_2} f[k_1,n_2]$$

Fast Fourier Transform

- is just the 1D DFT of column k_1 of $f[k_1,n_2]$



$f[k_1, n_2]$ $X[k_1, k_2]$

- this means that the 2D-DFT can be computed with a sequence of 1D-DFTs
- note that THIS DOES NOT REQUIRE SEPARABILITY
- this property is valid for any sequence
- it has obvious implications on the computational complexity of the DFT

Fast Fourier Transform

- note that the 2D-DFT requires
  - N_2 1D-DFTs on size N_1
  - followed by N_1 1D-DFTs of size N_2
  - when these are implemented with the FFT, total complexity is

$$
\begin{aligned}
&N_2 O(N_1 \log N_1) + N_1 O(N_2 \log N_2) = \\
&O(N_1 N_2 \log N_1) + O(N_1 N_2 \log N_2) = \\
&O(N_1 N_2 \log N_1 N_2)
\end{aligned}
$$

  - i.e. we have the same type of expression as in 1D
- in summary, the 2D-FFT simply consists of
  - 1) applying the 1D-FFT to the rows of the sequence
  - 2) applying the 1D-FFT to the columns of this intermediate sequence

Properties of the DFT

$x(n_1, n_2), y(n_1, n_2) = 0$ outside $0 \le n_1 \le N_1 - 1, 0 \le n_2 \le N_2 - 1$

$x(n_1, n_2) \longleftrightarrow X(k_1, k_2)$

$y(n_1, n_2) \longleftrightarrow Y(k_1, k_2)$

$N_1 \times N_2$-point DFT and IDFT are assumed.

Property 1. *Linearity*

$$ax(n_1, n_2) + by(n_1, n_2) \longleftrightarrow aX(k_1, k_2) + bY(k_1, k_2)$$

Property 2. *Circular Convolution*

$$x(n_1, n_2) \circledast y(n_1, n_2) \longleftrightarrow X(k_1, k_2)Y(k_1, k_2)$$

$$= [\tilde{x}(n_1, n_2) \circledast \tilde{y}(n_1, n_2)]R_{N_1 \times N_2}(n_1, n_2)$$

Property 3. *Relation between Circular and Linear Convolution*

$f(n_1, n_2) = 0$ outside $0 \le n_1 \le N_1' - 1, 0 \le n_2 \le N_2' - 1$

$g(n_1, n_2) = 0$ outside $0 \le n_1 \le N_1'' - 1, 0 \le n_2 \le N_2'' - 1$

$f(n_1, n_2) * g(n_1, n_2) = f(n_1, n_2) \circledast g(n_1, n_2)$ with periodicity $N_1 \ge N_1' + N_1'' - 1$, $N_2 \ge N_2' + N_2'' - 1$

Property 4. *Multiplication*

$$x(n_1, n_2)y(n_1, n_2) \longleftrightarrow \frac{1}{N_1N_2} X(k_1, k_2) \circledast Y(k_1, k_2)$$

$$= \frac{1}{N_1N_2} [\tilde{X}(k_1, k_2) \circledast \tilde{Y}(k_1, k_2)]R_{N_1 \times N_2}(k_1, k_2)$$

Properties of the DFT

Property 5. <u>*Separable Sequence*</u>

$$x(n_1, n_2) = x_1(n_1)x_2(n_2) \longleftrightarrow X(k_1, k_2) = X_1(k_1)X_2(k_2)$$

$X_1(k_1)$: N_1-point 1-D DFT

$X_2(k_2)$: N_2-point 1-D DFT

Property 6. <u>*Circular Shift of a Sequence*</u>

$$\tilde{x}(n_1 - m_1, n_2 - m_2)R_{N_1 \times N_2}(n_1, n_2) \longleftrightarrow X(k_1, k_2)e^{-j(2\pi/N_1)k_1m_1}e^{-j(2\pi/N_2)k_2m_2}$$

$$= x((n_1 - m_1)_{N_1}, (n_2 - m_2)_{N_2})$$

Property 7. <u>*Initial Value and DC Value Theorem*</u>

(a) $$x(0, 0) = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X(k_1, k_2)$$

(b) $$X(0, 0) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1, n_2)$$

Property 8. <u>*Parseval's Theorem*</u>

(a) $$\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1, n_2)y^*(n_1, n_2) = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X(k_1, k_2)Y^*(k_1, k_2)$$

(b) $$\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} |x(n_1, n_2)|^2 = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} |X(k_1, k_2)|^2$$

Properties of the DFT

Property 9. *Symmetry Properties*

(a) $x^*(n_1, n_2) \longleftrightarrow \tilde{X}^*(-k_1, -k_2)R_{N_1 \times N_2}(k_1, k_2) = X^*((-k_1)_{N_1}, (-k_2)_{N_2})$

(b) real $x(n_1, n_2) \longleftrightarrow$

$$X(k_1, k_2) = \tilde{X}^*(-k_1, -k_2)R_{N_1 \times N_2}(k_1, k_2)$$

$$X_R(k_1, k_2) = \tilde{X}_R(-k_1, -k_2)R_{N_1 \times N_2}(k_1, k_2)$$

$$X_I(k_1, k_2) = -\tilde{X}_I(-k_1, -k_2)R_{N_1 \times N_2}(k_1, k_2)$$

$$|X(k_1, k_2)| = |\tilde{X}(-k_1, -k_2)|R_{N_1 \times N_2}(k_1, k_2)$$

$$\theta_x(k_1, k_2) = -\tilde{\theta}_x(-k_1, -k_2)R_{N_1 \times N_2}(k_1, k_2)$$

Discrete Cosine Transform

- due to its computational efficiency the DFT is very popular
- however, it has strong disadvantages for some applications
  - it is complex
  - it has poor energy compaction
- energy compaction
  - is the ability to pack the energy of the spatial sequence into as few frequency coefficients as possible
  - this is very important for image compression
  - we represent the signal in the frequency domain
  - if compaction is high, we only have to transmit a few coefficients
  - instead of the whole set of pixels

Discrete Cosine Transform

- a much better transform, from this point of view, is the DCT
- it is defined by

$$C_x[k_1,k_2] = \begin{cases} \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x[n_1,n_2]\cos\left(\frac{\pi}{2N_1}k_1(2n_1+1)\right)\cos\left(\frac{\pi}{2N_2}k_2(2n_2+1)\right), & \begin{array}{l} 0 \le k_1 < N_1 \\ 0 \le k_2 < N_2 \end{array} \\ 0 & otherwise \end{cases}$$

$$x[n_1,n_2] = \begin{cases} \frac{1}{N_1 N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} w_1[k_1]w_2[k_2]C_x[k_1,k_2]\cos\left(\frac{\pi}{2N_1}k_1(2n_1+1)\right)\cos\left(\frac{\pi}{2N_2}k_2(2n_2+1)\right) & \begin{array}{l} 0 \le n_1 < N_1 \\ 0 \le n_2 < N_2 \end{array} \\ 0 & otherwise \end{cases}$$

with

$$w_1[k_1] = \begin{cases} 1/2, & k_1 = 0 \\ 1, & 1 \le k_1 < N_1 \end{cases}, \qquad w_2[k_2] = \begin{cases} 1/2, & k_2 = 0 \\ 1, & 1 \le k_2 < N_2 \end{cases}$$

- we will talk more about it in the next class

Any questions?