

Computation of the Multidimensional Discrete Fourier Transform (DFT)

Lecture by Prof. Brian L. Evans (UT Austin)

Scribe: Wade C. Schwartzkopf (UT Austin)

❖ DIRECT COMPUTATION OF TWO-DIMENSIONAL DFTS

◇ *If the twiddle factors (W_N terms) are pre-computed and stored in a table, then we require*

- $N_1^2 N_2^2$ complex multiplies and adds
- 10^{12} complex multiplies and adds (CMADs) if $N_1=N_2=1024$:

@1 $\mu\text{s}/\text{CMAD} \Rightarrow 11.5$ days (low-end programmable processor)

@1 ns/CMAD $\Rightarrow 16.7$ minutes (high-end dedicated processor)

Since bus speeds are significantly slower than processor speeds, we can only process the data in real time if we can feed data to the processor quickly enough. Note that a 200 MHz Texas Instruments TMS320C62x is a fixed-point programmable DSP processor that can compute one CMAD in 10 ns. A CMAD consists of *four* real multiplications plus *four* real additions, and the 200 MHz C62x can perform 400 million real multiplications per second. For more information about the C62x, see http://www.ece.utexas.edu/~bevans/courses/ee345s/lectures/01_Architecture

❖ ROW-COLUMN DECOMPOSITION

◇ *We can regroup the computations*

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \left[\sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_2}^{n_2 k_2} \right] W_{N_1}^{n_1 k_1} \quad \begin{array}{l} 0 \leq k_1 \leq N_1 - 1 \\ 0 \leq k_2 \leq N_2 - 1 \end{array}$$

$$G(n_1, k_2) = \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_2}^{n_2 k_2} \quad \begin{array}{l} 0 \leq n_1 \leq N_1 - 1 \\ 0 \leq k_2 \leq N_2 - 1 \end{array}$$

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} G(n_1, k_2) W_{N_1}^{n_1 k_1} \quad \begin{array}{l} 0 \leq k_1 \leq N_1 - 1 \\ 0 \leq k_2 \leq N_2 - 1 \end{array}$$

◇ **Interpretation**

- To compute $G(n, k_2)$, compute a 1-D DFT of the n^{th} column of x . Do this for all values of n .

◇ **Computation can be done using N_1 N_2 -point DFTs + N_2 N_1 -point DFT's**

- With direct computation of the 1-D DFTs, the requires

$$N_1 N_2^2 + N_2 N_1^2 = N_1 N_2 (N_1 + N_2) \text{ CMADS}$$

- 2×10^9 CMADs if $N_1 = N_2 = 1024$ for direct computation

$$@1 \mu\text{s/CMAD} \Rightarrow 33.3 \text{ minutes}$$

$$@1 \text{ ns/CMAD} \Rightarrow 2 \text{ seconds}$$

- If $N_1 = 2^u$; $N_2 = 2^v$ and a 2-D Fast Fourier Transform (FFT) is used

$$\frac{N_1 N_2}{2} \log_2 N_2 + \frac{N_1 N_2}{2} \log_2 N_1 = \frac{N_1 N_2}{2} \log_2 N_2 N_1$$

- 2×10^7 CMADs if $N_1 = N_2 = 1024$

$$@1 \mu\text{s/CMAD} \Rightarrow 10 \text{ seconds}$$

$$@1 \text{ ns/CMAD} \Rightarrow 0.01 \text{ seconds}$$

This execution time is now fast enough to perform a frame-based DFT for a video sequence at 30 frames/s.

- Note: Must be able to access data rowwise and columnwise. This needs memory!

❖ **MULTIDIMENSIONAL DFT'S**

◇ **Things become more complicated as the dimensionality goes up.**

$$X(k_1, k_2, \dots, k_M) = \sum_{n_1} \sum_{n_2} \dots \sum_{n_M} x(n_1, n_2, \dots, n_M) W_{N_M}^{n_M k_M} \dots W_{N_1}^{n_1 k_1}$$

1. Compute a 1-D DFT with regard to n_1 for each value of n_2, n_3, \dots, n_M .
2. Compute a 1-D DFT with regard to n_2 for each value of k_1, n_3, \dots, n_M .
- ...
- M. Compute a 1-D DFT with regard to n_M for each value of $k_1, k_2, \dots, k_{(M-1)}$

◇ Let $N_p = N_1 N_2 \dots N_M$, then

$$\begin{aligned} \# \text{CMADS} &= \sum_{i=1}^M \left(\prod_{\substack{j=1 \\ j \neq i}}^M N_j \right) \cdot \frac{N_i}{2} \log_2 N_i \\ &= \sum_{i=1}^M \underbrace{\left(\prod_{j=1}^M N_j \right)}_{N_p} \cdot \frac{1}{2} \log_2 N_i = N_p \cdot \frac{1}{2} \log_2 N_p \end{aligned}$$

❖ DEALING WITH STORAGE

◇ *If the machine on which an FFT is to be computed has limited memory, then it may be difficult to access the data both rowwise and columnwise.*

◇ *This is particularly true for 3-D and higher.*

◇ *In such cases the data is typically stored rowwise in secondary storage and matrix transposition is performed.*

1. Take row DFTs
 2. Transpose
 3. Take row DFTs
 4. Transpose
- etc.

❖ EKLUNDH'S TRANSPOSITION ALGORITHM

◇ *If $N_1 = N_2 = 2^u$, then transposition of a 2-D sequence (note that the origin is in the lower left-hand corner) can be performed recursively in u stages using only 2 rows of memory. The algorithm requires u passes through the matrix.*

$$X = \begin{bmatrix} X_2 & X_3 \\ X_0 & X_1 \end{bmatrix}$$

$$X^t = \begin{bmatrix} X_1^t & X_3^t \\ X_0^t & X_2^t \end{bmatrix}$$

❖ MORE GENERAL DFT ALGORITHMS

◇ *The rectangular DFT relates a finite extent rectangular array to rectangular samples of its Fourier transform.*

- Is there a DFT for hexagonal arrays?
- Is there a DFT connecting a rectangular array to hexagonal samples of its Fourier transform?

◇ **Definitions**

- $\tilde{x}(\mathbf{n})$: periodic sequence with periodicity matrix \mathbf{N} .
- J_N : set of $|\det(\mathbf{N})|$ samples that makes up exactly one period of $\tilde{x}(\mathbf{n})$.

◇ **Hypothesis**

- Let's hypothesize that $\tilde{x}(\mathbf{n})$ can be expressed as a finite linear combination of harmonically related complex sinusoids.

$$\tilde{x}(\mathbf{n}) = \sum_{\mathbf{k} \in J_N} a(\mathbf{k}) e^{j\mathbf{k}^T \mathbf{R}^T \mathbf{n}} \quad (*)$$

where \mathbf{k} is an integer vector and J_N is a set of multidimensional indices.

- If this representation is valid, it must be consistent with the periodicity of \tilde{x} .

$$\begin{aligned} \tilde{x}(\mathbf{n}) = \tilde{x}(\mathbf{n} + \mathbf{N}\mathbf{r}) &= \sum_{\mathbf{k} \in J_N} a(\mathbf{k}) e^{j\mathbf{k}^T \mathbf{R}^T (\mathbf{n} + \mathbf{N}\mathbf{r})} \\ &= \sum_{\mathbf{k} \in J_N} a(\mathbf{k}) e^{j\mathbf{k}^T \mathbf{R}^T \mathbf{N}\mathbf{r}} e^{j\mathbf{k}^T \mathbf{R}^T \mathbf{n}} \end{aligned} \quad (**)$$

- Equating the right sides of (*) and (**) and making use of the orthogonality property of the complex exponentials, for any harmonic that is not zero, we must have

$$e^{j\mathbf{k}^T \mathbf{R}^T \mathbf{N}\mathbf{r}} = 1$$

$$\mathbf{R}^T \mathbf{N} = 2\pi \mathbf{I}$$

$$\mathbf{R}^T = 2\pi \mathbf{N}^{-1}$$

- If we define $\tilde{X}(\mathbf{k}) = |\det \mathbf{N}| a(\mathbf{k})$, then

$$\tilde{x}(\mathbf{n}) = \frac{1}{|\det \mathbf{N}|} \sum_{\mathbf{k} \in J_{\mathbf{N}}} \tilde{X}(\mathbf{k}) e^{j\mathbf{k}^T (2\pi\mathbf{N}^{-1})\mathbf{n}}$$

- The complex exponentials in this sum are periodic in \mathbf{n} with periodicity matrix \mathbf{N} and in \mathbf{k} with periodicity matrix \mathbf{N}^t .

$\therefore J_{\mathbf{N}}$ contains $|\det \mathbf{N}^T| = |\det \mathbf{N}|$ members

$$\tilde{X}(\mathbf{k}) = \sum_{\mathbf{n} \in I_{\mathbf{N}}} \tilde{x}(\mathbf{n}) e^{-j\mathbf{k}^T (2\pi\mathbf{N}^{-1})\mathbf{n}}$$

- If we replace the periodic arrays \tilde{x} and \tilde{X} with finite support arrays \tilde{x} and \tilde{X} with support on $I_{\mathbf{N}}$ and $J_{\mathbf{N}}$, respectively, we obtain the generalized DFT.

$$x(\mathbf{n}) = \frac{1}{|\det \mathbf{N}|} \sum_{\mathbf{k} \in J_{\mathbf{N}}} X(\mathbf{k}) e^{j\mathbf{k}^T (2\pi\mathbf{N}^{-1})\mathbf{n}}, \quad \mathbf{n} \in I_{\mathbf{N}}$$

$$X(\mathbf{k}) = \sum_{\mathbf{n} \in I_{\mathbf{N}}} x(\mathbf{n}) e^{-j\mathbf{k}^T (2\pi\mathbf{N}^{-1})\mathbf{n}}, \quad \mathbf{k} \in J_{\mathbf{N}}$$

❖ SAMPLING THE FOURIER TRANSFORM

◇ **Define**

$$X(\boldsymbol{\omega}) = \sum_{\mathbf{n}} x(\mathbf{n}) e^{j\boldsymbol{\omega}^T \mathbf{n}}$$

◇ **Then**

$$X(\mathbf{k}) = X(\boldsymbol{\omega}) \Big|_{\boldsymbol{\omega}=(2\pi\mathbf{N}^{-T})\mathbf{k}}$$

◇ **Recall**

◇ **Note: If V is given, we have limited control over the locations of the**

$$\mathbf{\Omega} = \mathbf{V}^{-T} \mathbf{\omega}$$

$$\mathbf{\omega} = \mathbf{V}^T \mathbf{\Omega}$$

$$\mathbf{\Omega}_k = \mathbf{V}^{-T} (2\pi \mathbf{N}^{-T}) \mathbf{k} = 2\pi (\mathbf{V} \mathbf{N})^{-T} \mathbf{k}$$

frequency samples because of the constraint that N must be an integer matrix.

❖ QUINCUNX DFT

(Figure 3.10 in notes)

$$\mathbf{V} = \begin{bmatrix} \frac{T}{2} & \frac{T}{2} \\ \frac{T}{2} & -\frac{T}{2} \end{bmatrix}$$

◇ **If we use**

$$\mathbf{N} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix}$$

◇ **Then**

$$\Omega_k = \begin{bmatrix} \frac{1}{T} & \frac{1}{T} \\ \frac{1}{T} & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} \frac{2\pi}{N} & 0 \\ 0 & \frac{2\pi}{N} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \frac{2\pi}{NT} & \frac{2\pi}{NT} \\ \frac{2\pi}{NT} & -\frac{2\pi}{NT} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \frac{2\pi}{NT} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

◇ **This matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is called a quincunx sampling matrix (a non-rectangular sampling matrix of determinant $+2$ or -2). This is quincunx sampling.**

❖ EVALUATING GENERAL DFT'S

◇ **Fact: If \mathbf{N} is diagonal, the general DFT can be evaluated using a row-column algorithm**

$$\begin{aligned} X(k_1, k_2) &= \sum_{n_1} \sum_{n_2} x(n_1, n_2) e^{-j[k_1 \ k_2] \begin{bmatrix} \frac{2\pi}{N_1} & 0 \\ 0 & \frac{2\pi}{N_2} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}} \\ &= \sum_{n_1} \sum_{n_2} x(n_1, n_2) e^{-j\frac{2\pi n_1 k_1}{N_1}} e^{-j\frac{2\pi n_2 k_2}{N_2}} \end{aligned}$$

◇ **Smith Form Decomposition:**

- Any non-singular integer matrix can be expressed as

$$\mathbf{N} = \mathbf{U} \mathbf{D} \mathbf{V}$$

where

\mathbf{U}, \mathbf{V} : unimodular integer matrices (determinant of +1 or -1)

\mathbf{D} : diagonal integer matrix

Substituting a Smith form of a periodicity matrix \mathbf{N} ,

$$\begin{aligned} X(\mathbf{k}) &= \sum_{\mathbf{n} \in I_{\mathbf{N}}} x(\mathbf{n}) e^{-j\mathbf{k}^T (2\pi\mathbf{N}^{-1})\mathbf{n}} \\ &= \sum_{\mathbf{n} \in I_{\mathbf{N}}} x(\mathbf{n}) e^{-j\mathbf{k}^T (2\pi\mathbf{V}^{-1}\mathbf{D}^{-1}\mathbf{U}^{-1})\mathbf{n}} \\ &= \sum_{\mathbf{n} \in I_{\mathbf{N}}} x(\mathbf{n}) e^{-j2\pi(\mathbf{V}^{-T}\mathbf{k})^T \mathbf{D}^{-1}(\mathbf{U}^{-1}\mathbf{n})} \end{aligned}$$

We would like to express the DFT of a signal with non-rectangular periodicity in terms of a separable DFT. We would like the term in the

$$-j2\pi \hat{\mathbf{k}}^T \mathbf{D}^{-1} \hat{\mathbf{n}}$$

exponential to be of the form

We work towards this goal by making the following change of variables:

◇ **Algorithm**

$$\text{Let } \hat{\mathbf{k}} = \mathbf{V}^{-T}\mathbf{k} \quad \hat{\mathbf{n}} = \mathbf{U}^{-1}\mathbf{n}$$

$$\mathbf{k} = \mathbf{V}^T \hat{\mathbf{k}} \quad \mathbf{n} = \mathbf{U} \hat{\mathbf{n}}$$

$$X(\mathbf{V}^T \hat{\mathbf{k}}) = \sum_{\mathbf{U}\hat{\mathbf{n}} \in I_{\mathbf{N}}} x(\mathbf{U} \hat{\mathbf{n}}) e^{-j2\pi \hat{\mathbf{k}}^T \mathbf{D}^{-1} \hat{\mathbf{n}}}$$

1. Compute $x(\hat{\mathbf{n}})$ by permuting $x(\mathbf{n})$ according to $\hat{\mathbf{n}} = \mathbf{U}^{-1} \mathbf{n}$. The permuted values should fall (or be made to fall using circular shifting) into the fundamental parallelepiped formed by \mathbf{D} , i.e. a rectangular volume of samples.
2. Calculate a separable DFT according to entries on the diagonals of \mathbf{D}
3. Permute DFT $X(\hat{\mathbf{k}})$ using $\mathbf{k} = \mathbf{V}^T \hat{\mathbf{k}}$.