Fast Transform Algorithms via Projection on a Subalgebra

Yevgen Voronenko Frédéric de Mesmay

Electrical and Computer Engineering
Carnegie Mellon University

Introduction

- Fast *n log n* algorithms are important for practical applications
- Deriving such algorithms is typically tedious and not well understood process
- We present a general method for mechanical derivation of new algorithms by *projecting* existing algorithms
- We successfully obtain fast algorithms for all 16 types of DCTs/DSTs

Related Work : Trigonometric Transform Algorithms

- Hong and Vetterli, 1991
  - "Basefield Transforms with Convolution Property"
  - Algebraic projection (by human)
- Frigo and Johnson, 1999
  - "Fast Fourier Transform Compiler"
  - Data-flow graph projection (by a computer)
- Pueschel and Egner, 2002
  - Symmetry Based Matrix Factorization
  - Automatic algorithm derivation from a matrix (by a computer)
- Pueschel and Moura, 2001—2006
  - Algebraic Theory of Signal Processing
  - Cooley-Tukey type algorithms for all 16 kinds of DCTs/DSTs

ASP allows to understand what's going on and thus mechanically derive algorithms

Outline

1. Introduction
2. Embedding method
3. Algorithm projection method

Embedding Method: Problem statement

- Given:
  - Regular signal model $(\mathcal{A}, \Phi_A)$ + Fourier transform $\mathcal{F}_A$
  - Regular signal model $(\mathcal{B}, \Phi_B)$ + Fourier transform $\mathcal{F}_B$
  - $\mathcal{B} \leq \mathcal{A}$

- Find:
  - Express $\mathcal{F}_B$ using $\mathcal{F}_A$

Embedding Method

- Basic idea: embed $\mathcal{B}$ in $\mathcal{A}$
- Let ϕ and Ψ be embeddings

$$\phi : \mathcal{B} \rightarrow \mathcal{A} \qquad\qquad \psi : \bigoplus \mathcal{B}_i \rightarrow \bigoplus \mathcal{A}_i$$

- Following diagram commutes

$$\begin{tikzcd}
\mathcal{B} \arrow[r, "\phi"] \arrow[d, "\mathcal{F}_B"'] & \mathcal{A} \arrow[d, "\mathcal{F}_A"] \\
\bigoplus \mathcal{B}_i & \bigoplus \mathcal{A}_i \arrow[l, "\psi^{-1}"']
\end{tikzcd}$$

What we want

What we have

Matrix equation

$$\mathcal{F}_B = \underbrace{C}_{\text{coordinatized } \Psi^{-1}} \cdot \mathcal{F}_A \cdot \underbrace{E}_{\text{coordinatized } \phi}$$

Challenge: coordinatize mappings ϕ and Ψ^{-1}

Example: DFT

- Consider $\mathcal{A} = \mathbb{C}[x]/x^8 - 1$

$$\mathcal{B} = < x^2 > = \mathbb{C}[y]/y^4 - 1, \quad y = x^2$$

- We have

$$\begin{tikzcd}
\mathbb{C}[y]/y^4 - 1 \arrow[r, "\phi", "E"'] \arrow[d, "\mathrm{DFT}_4"'] & \mathbb{C}[x]/x^8 - 1 \arrow[d, "\mathrm{DFT}_8"] \\
\bigoplus \mathbb{C}[y]/(y - \omega_4^i) & \bigoplus \mathbb{C}[x]/(x - \omega_8^i) \arrow[l, "\psi^{-1}"', "C"]
\end{tikzcd}$$

$$E = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \qquad C = 1/2 \begin{pmatrix} 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}$$

$$\boxed{\mathrm{DFT}_4 = C \cdot \mathrm{DFT}_8 \cdot E}$$

$$E_2 = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Crucial parameters: basis in $\mathcal{A}$, $\mathcal{B}$, $\mathcal{A}_i$, $\mathcal{B}_i$

Multiple degrees of freedom

Example: DCT-3

- Consider $\mathcal{A} = \mathbb{R}[x]/x^8 + 1$

$$\mathcal{B} = \mathbb{R}[y]/T_4(y), \quad y = (x + x^{-1})/2$$

- We have

$$\begin{array}{ccc} \mathbb{R}[y]/T_4(y) & \xrightarrow[E]{\phi} & \mathbb{R}[x]/x^8 + 1 \\ \text{DCT-3}_4 \big\downarrow & & \big\downarrow \text{RDFT-3}_8 \\ \bigoplus \mathbb{C}[y]/(y - \cos \alpha_i) & \xleftarrow[C]{\psi^{-1}} & \bigoplus \mathbb{C}[x]/(x - \omega_8^i)(x - \omega_8^{8-i}) \end{array}$$

$$E = W_A = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1/2 & \cdot & \cdot \\ \cdot & \cdot & 1/2 & \cdot \\ \cdot & \cdot & \cdot & 1/2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & -1/2 \\ \cdot & \cdot & -1/2 & \\ \cdot & -1/2 & & \cdot \end{pmatrix}$$

$$C = DS_1 = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}$$

$$\text{DCT-3}_4 = DS_1 \cdot \text{RDFT-3}_8 \cdot W_A$$

$$\text{DST-3}_4 = DS_2 \cdot \text{RDFT-3}_8 \cdot W_{S0}$$

Outline

1. Introduction
2. Embedding method
3. Algorithm Embedding method.

Projection Method: Problem statement

- Given:
  - Regular signal model $(\mathcal{A}, \Phi_A)$ + Fourier transform $\mathcal{F}_A$
  - Cooley-Tukey type algorithm for $\mathcal{F}_A$ made with sub-algebra $\mathcal{C}$

  - Regular signal model $(\mathcal{B}, \Phi_B)$ + Fourier transform $\mathcal{F}_B$
  - $\mathcal{B} \leq \mathcal{A}$

- Find:
  - Cooley-Tukey type algorithm for $\mathcal{F}_B$

Reminder: Cooley-Tukey Type Algorithms

Assume

$p(x) = q(r(x))$

$$\begin{tikzcd}
\mathbb{C}[x]/p(x) \arrow[dd, "\text{transform } T"'] \arrow[dr, "\text{coarse decomposition} \quad (\mathcal{F}' \otimes I)B"] & \\
& \mathbb{C}[x]/(r(x)-\beta_1) \oplus \ldots \oplus \mathbb{C}[x]/(r(x)-\beta_n) \arrow[dl, "\text{complete decomposition} \quad P(\bigoplus \mathcal{F}_i)"] \\
\mathbb{C}[x]/(x-\alpha_1) \oplus \ldots \oplus \mathbb{C}[x]/(x-\alpha_n) &
\end{tikzcd}$$

Fast Algorithms DFT and RDFT

Decomposition: $x^n - 1 = (x^m)^k - 1$

$$\begin{tikzcd}
\mathbb{C}[x]/x^n-1 \arrow[dd, "\mathrm{DFT}_n"'] \arrow[dr, "(\mathrm{DFT}_k \otimes I_m)"] & \\
& \bigoplus \mathbb{C}[x]/(x^m-\omega_k^i) \arrow[dl, "L(\bigoplus \mathrm{DFT}(\omega_k^i))"] \\
\bigoplus \mathbb{C}[x]/(x-\omega_n^i) &
\end{tikzcd}$$

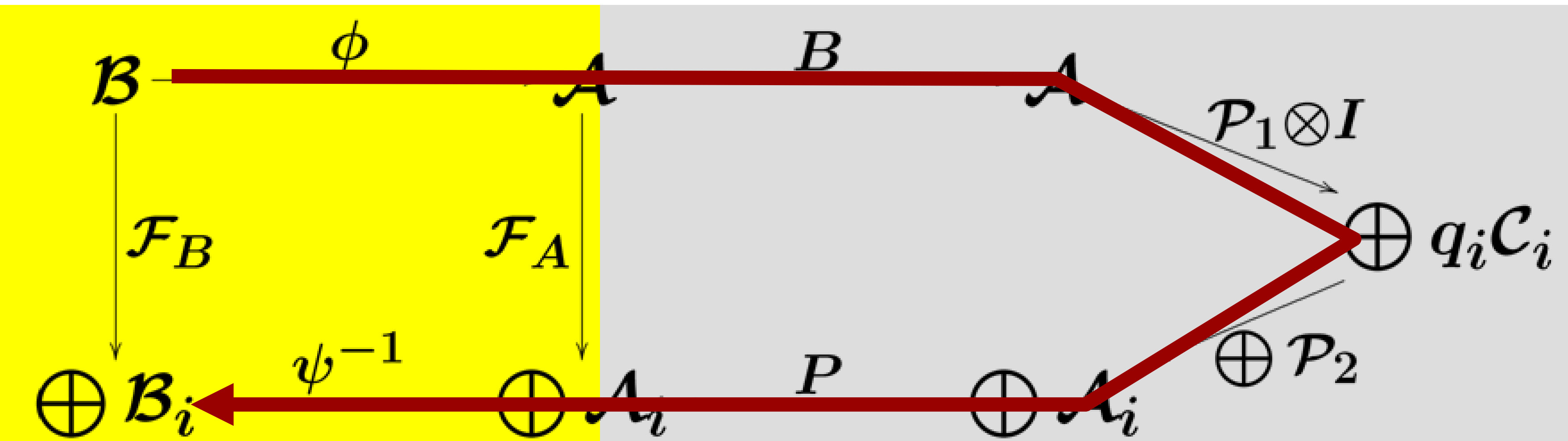
$$\begin{tikzcd}
\mathbb{R}[x]/x^n-1 \arrow[dd, "\mathrm{RDFT}_n"'] \arrow[dr, "(\overline{\mathrm{RDFT}}_k \otimes I_m)"] & \\
& \bigoplus \mathbb{R}[x]/x^{2m}-2x^m\cos\alpha+1 \arrow[dl, "\overline{K}(\bigoplus \mathrm{RDFT}(\alpha_i))"] \\
\bigoplus \mathbb{R}[x]/x^2-2x\cos\alpha_i+1 &
\end{tikzcd}$$

$$\mathrm{DFT}_n = \underline{L_m^n(I_k \otimes \mathrm{DFT}_m)T_m^n}\,\underline{(\mathrm{DFT}_k \otimes I_m)}$$

$$\mathrm{RDFT}_n = \underline{(K_{m/2}^{n/2} \otimes I_2)(\bigoplus \mathrm{RDFT}_m(\alpha_i))}\,\underline{(\overline{\mathrm{RDFT}}_k \otimes I_m)}$$

Algorithm Projection Method

- Basic idea: embedding and Cooley-Tukey
- Following diagram commutes



The embedding　　　　The $\mathcal{F}_A$ Cooley-Tukey

Matrix equation $\mathcal{F}_B = C \cdot P \cdot (\bigoplus \mathcal{P}_2) \cdot (\mathcal{P}_1 \otimes I) \cdot B \cdot E$

coordinatized ψ　spectral permutation　fine decomposition　coarse decomposition　base change　coordinatized ϕ

Challenge: choose ϕ and Ψ^{-1} for sparse factorization

Algorithm Projection Steps

- Algorithm derivation is mechanical rewriting of

$$\mathcal{F}_B \;\rightarrow\; \underbrace{\mathrm{C}\cdot P(\bigoplus \mathcal{P}_2(\beta_i))}_{\text{apply left rules}}\underbrace{(\mathcal{P}_1 \otimes \mathrm{I})B\cdot \mathrm{E}}_{\text{apply right rules}}$$

$$\mathcal{F}_B = \mathrm{C}\cdot\mathcal{F}_A\cdot\mathrm{E}$$

$$\mathrm{C}\cdot\mathcal{F}_A \;\rightarrow\; \mathrm{E}^{-1}\cdot\mathcal{F}_B \qquad\qquad \mathcal{F}_A\cdot\mathrm{E} \;\rightarrow\; \mathrm{C}^{-1}\cdot\mathcal{F}_B$$

left “push” rules — right “pull” rules

- Step 1: Write down the embedding
- Step 2: Coordinatize and obtain right / left rules
- Step 3: Plug in $\mathcal{F}_A$ Cooley-Tukey and apply rules

But remember that E and C are chosen!

Example: Embedding DFT_4 into DFT_8

- Step 1: Write down the embedding

$$\mathcal{A} = \mathbb{C}[x]/x^8 - 1$$

$$\mathcal{B} = \mathbb{C}[y]/y^4 - 1 =< x^2 >$$

- Step 2: Coordinatize, and obtain right/left rules

$$C_n \cdot \mathrm{DFT}_n \cdot E_n = \mathrm{DFT}_{n/2},$$

$$\mathrm{DFT}_n \, E_n = C_n^{-1} \, \mathrm{DFT}_{n/2},$$

$$C_n \, \mathrm{DFT}_n = \mathrm{DFT}_{n/2} \, E_n^{-1}.$$

$$C_8 = 1/2 \begin{pmatrix} 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \end{pmatrix}$$

$$E_8 = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

- Step 3: Plug in Cooley-Tukey for DFT_8, apply rules

$$\mathrm{DFT}_4 = C_8 \cdot L^8_4 (\oplus \mathrm{DFT}_4(r_i))(\mathrm{DFT}_2 \otimes I_4) \cdot E_8$$

$$\mathrm{DFT}_4 = C_8 \cdot L^8_4 (\oplus \mathrm{DFT}_4(r_i)) E_8 (\mathrm{DFT}_2 \otimes I_2)$$

$$\mathrm{DFT}_4 = L^8_4 (\oplus C_4 \mathrm{DFT}_4(r_i)) E_8 (\mathrm{DFT}_2 \otimes I_2)$$

DF

$$\mathrm{DFT}_4 = L^8_4 (\oplus \mathrm{DFT}_2(r_i))(\mathrm{DFT}_2 \otimes I_2)$$

Example: Embed DCT-3_4 into RDFT-3_8

- Step 1: Write down the embedding

$$\mathcal{A} = \mathbb{R}[x]/x^8 + 1$$

$$\mathcal{B} = \mathbb{R}[y]/T_4(y) = \left\langle (x + x^{-1})/2 \right\rangle$$

- Step 2: Coordinatize, and obtain right/left rules

$$\begin{aligned} C_n \cdot \text{RDFT-3}_n \cdot E_n &= \text{DCT-3}_{n/2}, \\ \text{RDFT-3}_n\, E_n &= C_n^{-1}\, \text{DCT-3}_{n/2}, \\ C_n\, \text{RDFT-3}_n &= \text{DCT-3}_{n/2}\, E_n^{-1}. \end{aligned}$$

$$C_8 = \begin{pmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}$$

$$E_8 = \frac{1}{2}\begin{pmatrix} 2 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & -1 \\ \cdot & \cdot & -1 & \cdot \\ \cdot & -1 & \cdot & \cdot \end{pmatrix}$$

- Step 3: Plug in Cooley-Tukey for RDFT-3_8, apply rules

$$\text{DCT-3}_4 = C_8 \cdot (K_2^4 \otimes I_2)\Big(\bigoplus \text{RDFT-3}_4(r_i)\Big)(\overline{\text{RDFT-3}_4 \otimes I_2}) \cdot E_8$$

$$\text{DCT-3}_4 = K_2^4\Big(\bigoplus \text{DCT-3}_2(r_i)\Big)Q(\overline{\text{DCT-3}} \oplus (I \otimes \text{RDFT-3}))G(\oplus\overline{\text{DCT-4}})P$$

permutations

Conclusion : Automatic Derivation of Algorithms

$$\mathrm{RDFT}_n = (K^{n/2}_{m/2} \otimes I_2)\left(\bigoplus \mathrm{RDFT}_m(\alpha_i)\right)(\overline{\mathrm{RDFT}}_k \otimes I_m)$$

Mechanical formula rewriting

rewrite rules

$$\mathrm{DCT\text{-}3}_k = K^{mn}_m\left(\bigoplus \mathrm{DCT\text{-}3}_m(2r_i)\right)Q(\overline{\mathrm{DCT\text{-}3}}_n \oplus (\mathrm{I}_{m/2-1} \otimes \mathrm{RDFT\text{-}3}_{2n})G \oplus \overline{\mathrm{DCT\text{-}4}}_n)R(L^{2nm}_m)$$

- Number of rewrite rules is small
- Hard to derive by hand
- Gives new, composite algorithms
- Nice way to handle the ever growing number of transforms

Questions?