ITCT Lecture 10.4:
Fast Algorithm for the DCT

Prof. Ja-Ling Wu

Department of Computer Science
and Information Engineering
National Taiwan University

1

Fast Algorithm for the DCT

- The DCT matrix is orthogonal; its inverse is its transpose.

Definition : $\boxed{\text{N - point DCT}}$

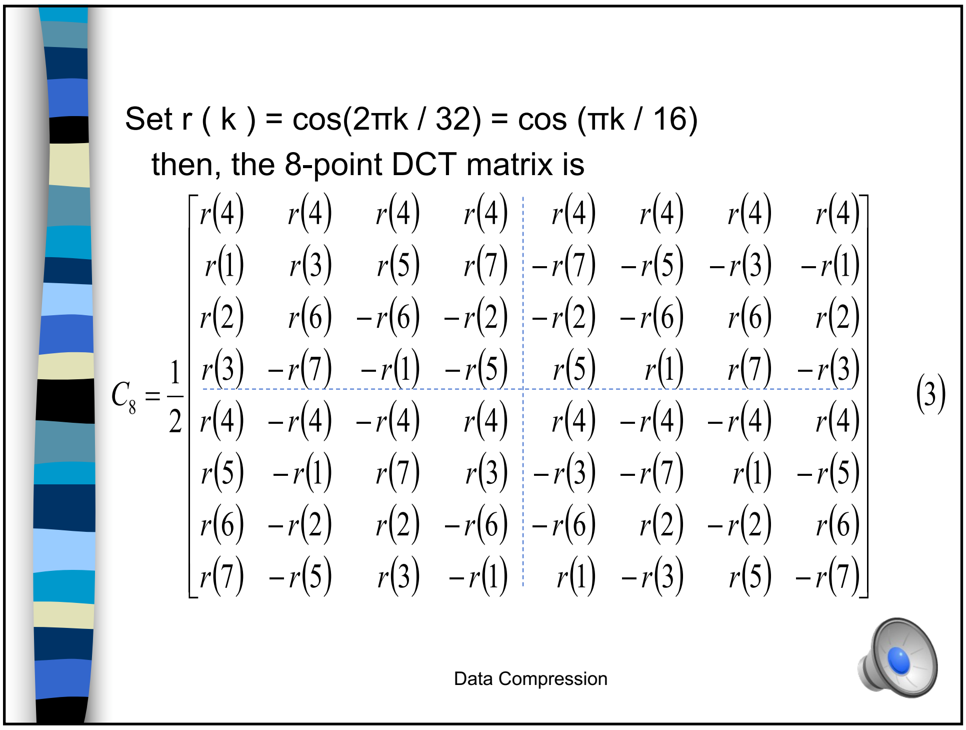
$$y_n = C_n \sum_{k=0}^{N-1} x_k \cos\frac{2\pi n(2k+1)}{4N} \qquad (1)$$

$\boxed{\text{N – point IDCT}}$

$$x_k = \sum_{n=0}^{N-1} C_n y_n \cos\frac{2\pi n(2k+1)}{4N} \qquad (2)$$

Let us focus on the case of N = 8.

2



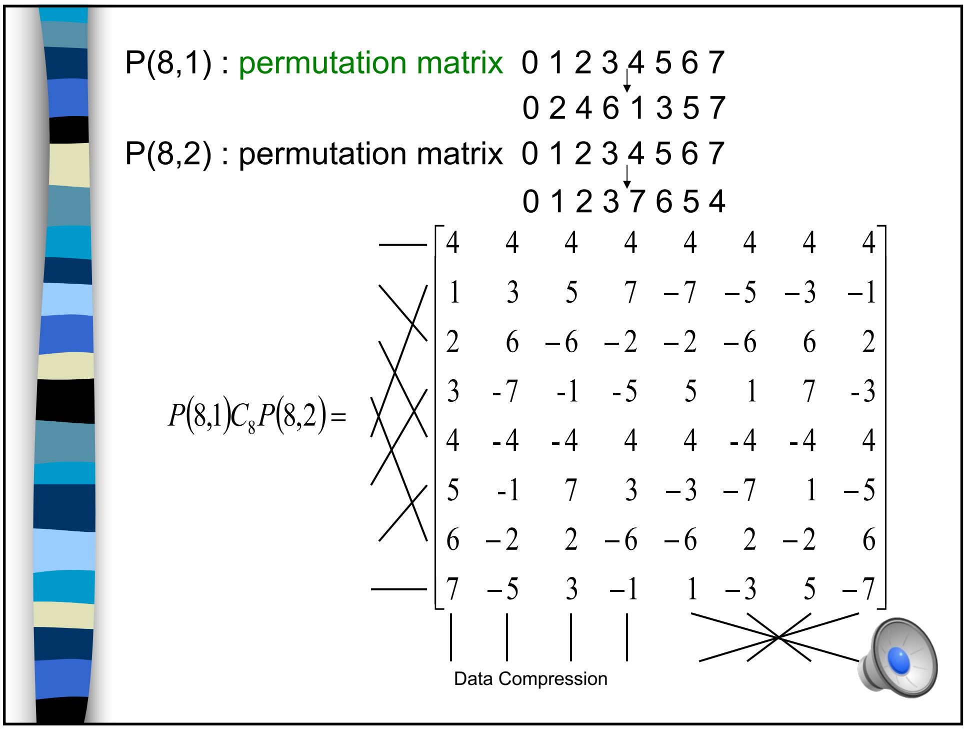
Set r (k) = cos(2πk / 32) = cos (πk / 16)

then, the 8-point DCT matrix is

$$C_8 = \frac{1}{2}\begin{bmatrix} r(4) & r(4) & r(4) & r(4) & r(4) & r(4) & r(4) & r(4) \\ r(1) & r(3) & r(5) & r(7) & -r(7) & -r(5) & -r(3) & -r(1) \\ r(2) & r(6) & -r(6) & -r(2) & -r(2) & -r(6) & r(6) & r(2) \\ r(3) & -r(7) & -r(1) & -r(5) & r(5) & r(1) & r(7) & -r(3) \\ r(4) & -r(4) & -r(4) & r(4) & r(4) & -r(4) & -r(4) & r(4) \\ r(5) & -r(1) & r(7) & r(3) & -r(3) & -r(7) & r(1) & -r(5) \\ r(6) & -r(2) & r(2) & -r(6) & -r(6) & r(2) & -r(2) & r(6) \\ r(7) & -r(5) & r(3) & -r(1) & r(1) & -r(3) & r(5) & -r(7) \end{bmatrix} \qquad (3)$$

Data Compression

3



P(8,1) : permutation matrix 0 1 2 3 4 5 6 7 ↓ 0 2 4 6 1 3 5 7

P(8,2) : permutation matrix 0 1 2 3 4 5 6 7 ↓ 0 1 2 3 7 6 5 4

$$P(8,1)C_8P(8,2)=\begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 3 & 5 & 7 & -7 & -5 & -3 & -1 \\ 2 & 6 & -6 & -2 & -2 & -6 & 6 & 2 \\ 3 & -7 & -1 & -5 & 5 & 1 & 7 & -3 \\ 4 & -4 & -4 & 4 & 4 & -4 & -4 & 4 \\ 5 & -1 & 7 & 3 & -3 & -7 & 1 & -5 \\ 6 & -2 & 2 & -6 & -6 & 2 & -2 & 6 \\ 7 & -5 & 3 & -1 & 1 & -3 & 5 & -7 \end{bmatrix}$$

Data Compression

4

$$
=\left[\begin{array}{cccc|cccc}
4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
2 & 6 & -6 & -2 & 2 & 6 & -6 & -2 \\
4 & -4 & -4 & 4 & 4 & -4 & -4 & 4 \\
6 & -2 & 2 & -6 & 6 & -2 & 2 & -6 \\
\hline
1 & 3 & 5 & 7 & -1 & -3 & -5 & -7 \\
3 & -7 & -1 & -5 & -3 & 7 & 1 & 5 \\
5 & -1 & 7 & 3 & -5 & 1 & -7 & -3 \\
7 & -5 & 3 & -1 & -7 & 5 & -3 & 1
\end{array}\right]
\qquad
\begin{bmatrix} A & A \\ B & -B \end{bmatrix}
=\left[\begin{array}{c|c} A & \\ \hline & B \end{array}\right]
\otimes
\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}
$$

Let I_n denote the n×n identity matrix,

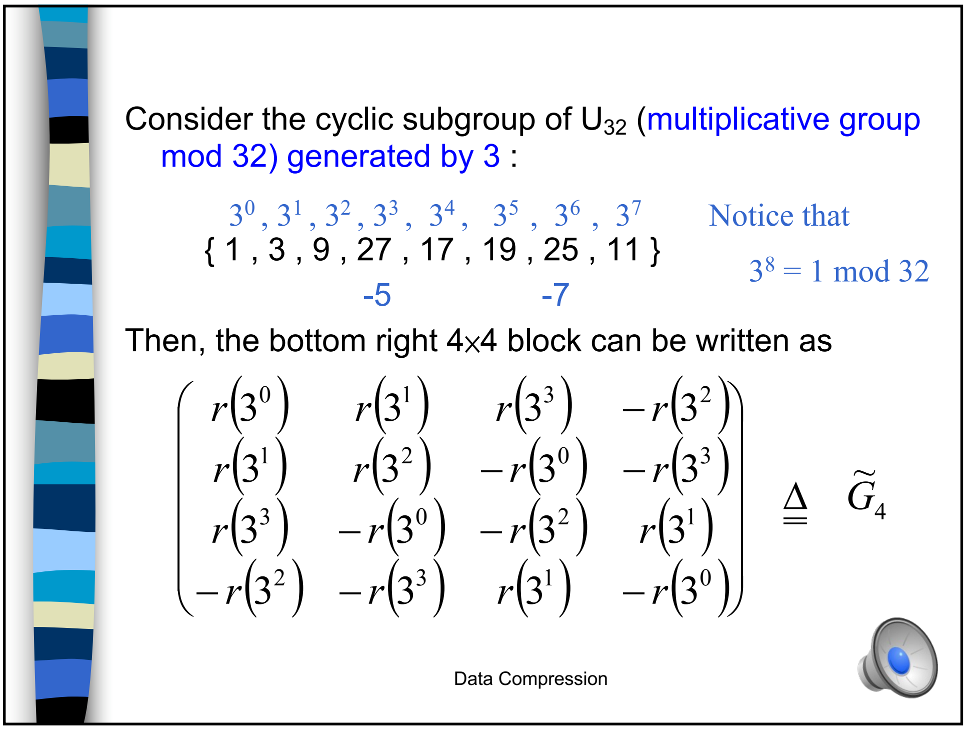
$$F \triangleq \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

and define $R_8 = F \otimes I_4$

(if $A=(a_{i,\,j})_{m\times n}$, $B=(b_{i,\,j})_{p\times g}$, then the tensor product $A \otimes B$ is the mp×ng matrix composed of the m×n blocks ($a_{i,j}$ B))

Then

$$
P(8,1)C_8P(8,2)R_8 =
\begin{bmatrix}
r(4) & r(4) & r(4) & r(4) & & & & \\
r(2) & r(6) & -r(6) & -r(2) & & & & \\
r(4) & -r(4) & -r(4) & r(4) & & & & \\
r(6) & -r(2) & r(2) & -r(6) & & & & \\
 & & & & r(1) & r(3) & r(5) & r(7) \\
 & & & & r(3) & -r(7) & -r(1) & -r(5) \\
 & & & & r(5) & -r(1) & r(7) & r(3) \\
 & & & & r(7) & -r(5) & r(3) & -r(1)
\end{bmatrix}
\tag{4}
$$



Consider the cyclic subgroup of U_{32} (multiplicative group mod 32) generated by 3 :

$$\begin{array}{cccccccc} 3^0 , & 3^1 , & 3^2 , & 3^3 , & 3^4 , & 3^5 , & 3^6 , & 3^7 \\ \{\, 1 , & 3 , & 9 , & 27 , & 17 , & 19 , & 25 , & 11 \,\} \\ & & & -5 & & & -7 & \end{array}$$

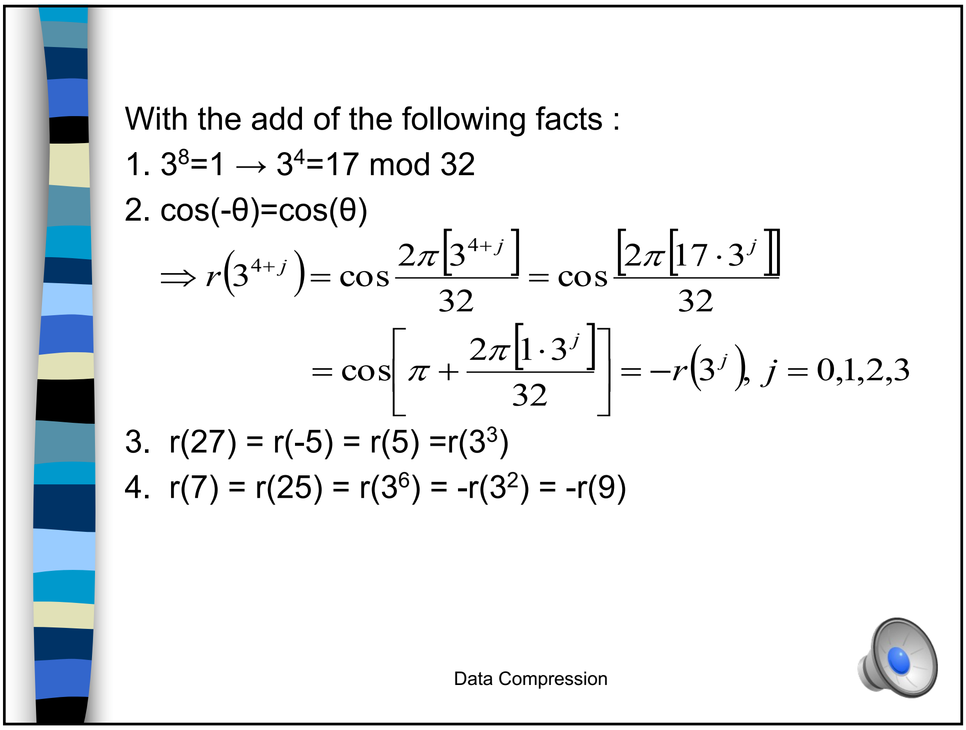
Notice that

$3^8 = 1 \bmod 32$

Then, the bottom right 4×4 block can be written as

$$\begin{pmatrix} r(3^0) & r(3^1) & r(3^3) & -r(3^2) \\ r(3^1) & r(3^2) & -r(3^0) & -r(3^3) \\ r(3^3) & -r(3^0) & -r(3^2) & r(3^1) \\ -r(3^2) & -r(3^3) & r(3^1) & -r(3^0) \end{pmatrix} \stackrel{\Delta}{=} \widetilde{G}_4$$

7



With the add of the following facts :

1. $3^8=1 \rightarrow 3^4=17 \bmod 32$
2. $\cos(-\theta)=\cos(\theta)$

$$\Rightarrow r\left(3^{4+j}\right) = \cos\frac{2\pi\left[3^{4+j}\right]}{32} = \cos\frac{\left[2\pi\left[17\cdot 3^j\right]\right]}{32}$$

$$= \cos\left[\pi + \frac{2\pi\left[1\cdot 3^j\right]}{32}\right] = -r\left(3^j\right),\ j = 0,1,2,3$$

3. $r(27) = r(-5) = r(5) = r(3^3)$
4. $r(7) = r(25) = r(3^6) = -r(3^2) = -r(9)$

8

Define

$$P(4,1)=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&-1\\0&0&1&0\end{pmatrix}$$

$$P(4,1)\begin{pmatrix}r(1)\\r(3)\\r(5)\\r(7)\end{pmatrix}=\begin{pmatrix}r(1)\\r(3)\\-r(7)\\r(5)\end{pmatrix}=\begin{pmatrix}r(3^0)\\r(3^1)\\r(3^2)\\r(3^3)\end{pmatrix}$$

Then,

$$P(4,1)\widetilde{G}_4P(4,1)^{-1}=\begin{pmatrix}r(3^0)&r(3^1)&r(3^2)&r(3^3)\\r(3^1)&r(3^2)&r(3^3)&-r(3^0)\\r(3^2)&r(3^3)&-r(3^0)&-r(3^1)\\r(3^3)&-r(3^0)&-r(3^1)&-r(3^2)\end{pmatrix}$$

: signed-circulant matrix

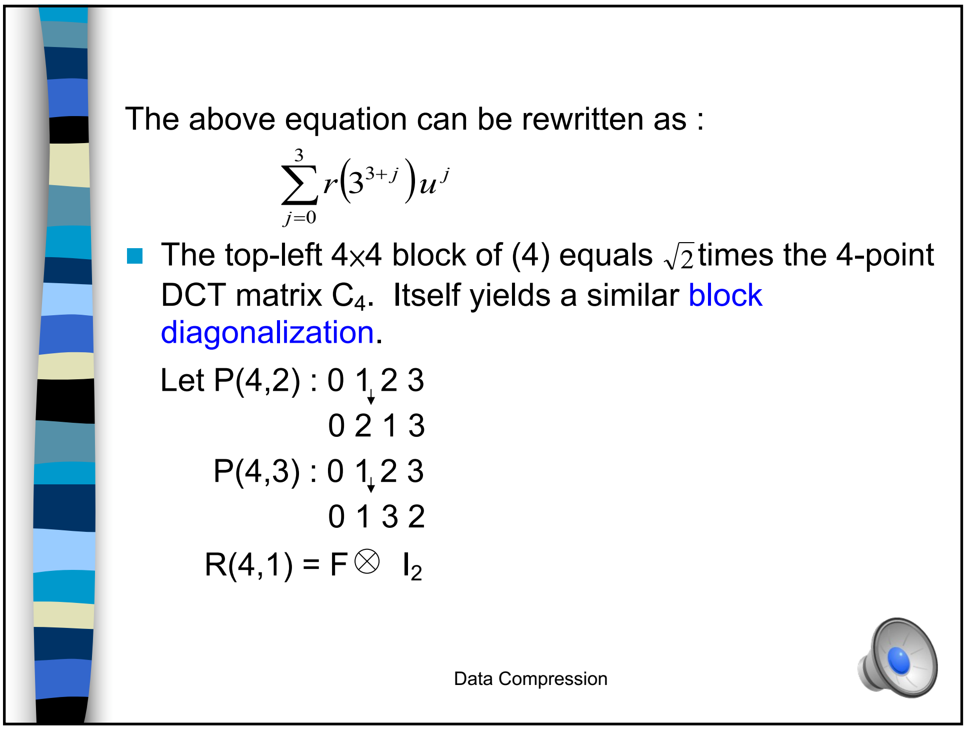
Reversing the order of the columns in $P(4,1)\widetilde{G}_4P(4,1)^{-1}$ yields

$$G_4=\begin{pmatrix}r(3^3)&r(3^2)&r(3^1)&r(3^0)\\-r(3^0)&r(3^3)&r(3^2)&r(3^1)\\-r(3^1)&-r(3^0)&r(3^3)&r(3^2)\\-r(3^2)&-r(3^1)&-r(3^0)&r(3^3)\end{pmatrix}\qquad(5)$$

G4 can be viewed as an element in the regular representation of the polynomial ring in the variable u modulo (u^4+1).

$$\begin{pmatrix}w_0\\w_1\\w_2\\w_3\end{pmatrix}=\begin{pmatrix}r(3^3)&r(3^2)&r(3^1)&r(3^0)\\-r(3^0)&r(3^3)&r(3^2)&r(3^1)\\-r(3^1)&-r(3^0)&r(3^3)&r(3^2)\\-r(3^2)&-r(3^1)&-r(3^0)&r(3^3)\end{pmatrix}\begin{pmatrix}v_0\\v_1\\v_2\\v_3\end{pmatrix},\quad\text{then}$$

$$\left(r(3^3)-r(3^0)u-r(3^1)u^2-r(3^2)u^3\right)\cdot\left(v_0+v_1u+v_2u^2+v_3u^3\right)\quad(6)$$
$$=\left(w_0+w_1u+w_2u^2+w_3u^3\right)\quad\text{mod}\quad\left(u^4+1\right)$$



The above equation can be rewritten as :

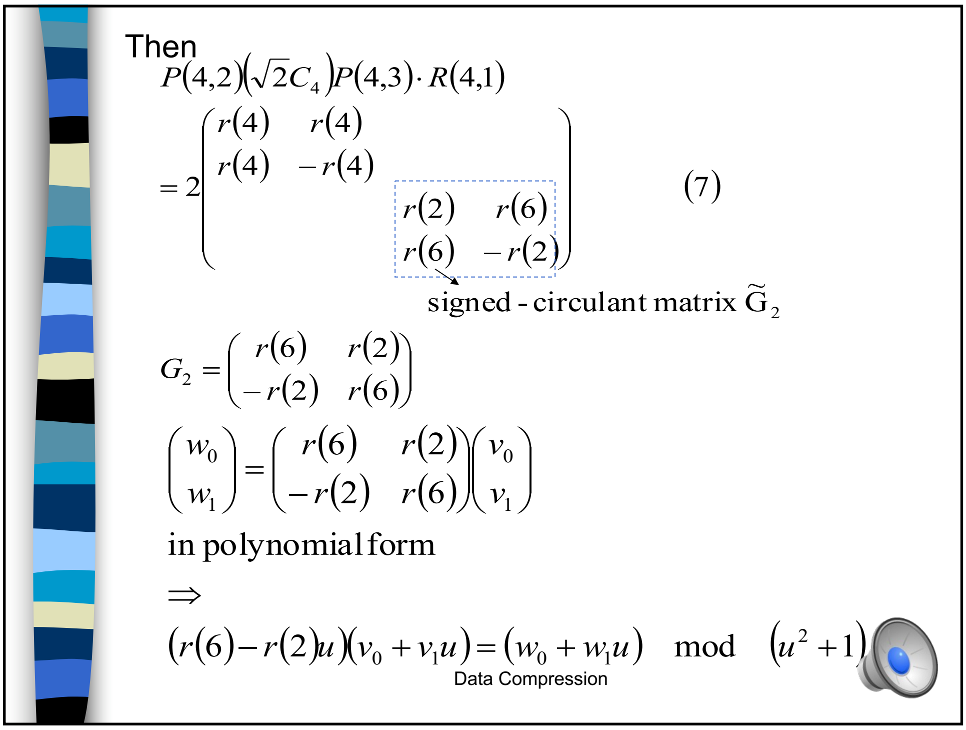
$$\sum_{j=0}^{3} r\left(3^{3+j}\right)u^{j}$$

- The top-left 4x4 block of (4) equals $\sqrt{2}$ times the 4-point DCT matrix C_4. Itself yields a similar block diagonalization.

Let P(4,2) : 0 1 2 3
0 2 1 3

P(4,3) : 0 1 2 3
0 1 3 2

R(4,1) = F ⊗ I_2



Then

$$P(4,2)\left(\sqrt{2}C_4\right)P(4,3)\cdot R(4,1)$$

$$=2\begin{pmatrix} r(4) & r(4) & & \\ r(4) & -r(4) & & \\ & & \boxed{\begin{matrix} r(2) & r(6) \\ r(6) & -r(2) \end{matrix}} & \end{pmatrix} \qquad (7)$$

signed - circulant matrix $\tilde{G}_2$

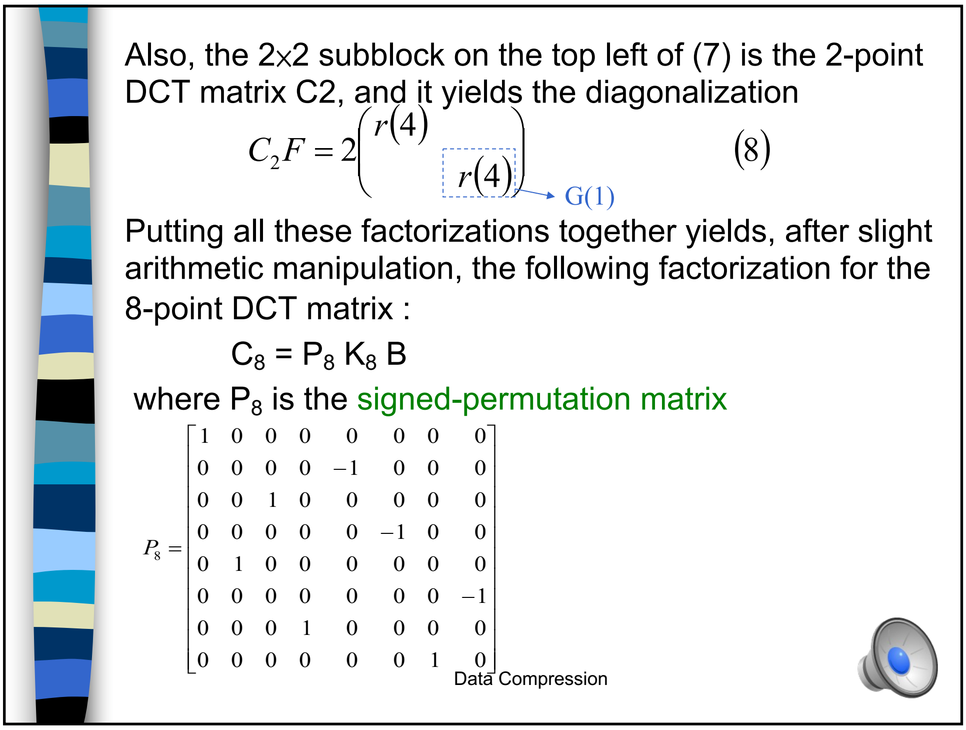
$$G_2=\begin{pmatrix} r(6) & r(2) \\ -r(2) & r(6) \end{pmatrix}$$

$$\begin{pmatrix} w_0 \\ w_1 \end{pmatrix}=\begin{pmatrix} r(6) & r(2) \\ -r(2) & r(6) \end{pmatrix}\begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$$

in polynomial form

$\Rightarrow$

$$\left(r(6)-r(2)u\right)\left(v_0+v_1u\right)=\left(w_0+w_1u\right) \quad \text{mod} \quad \left(u^2+1\right)$$



Also, the 2×2 subblock on the top left of (7) is the 2-point DCT matrix C2, and it yields the diagonalization

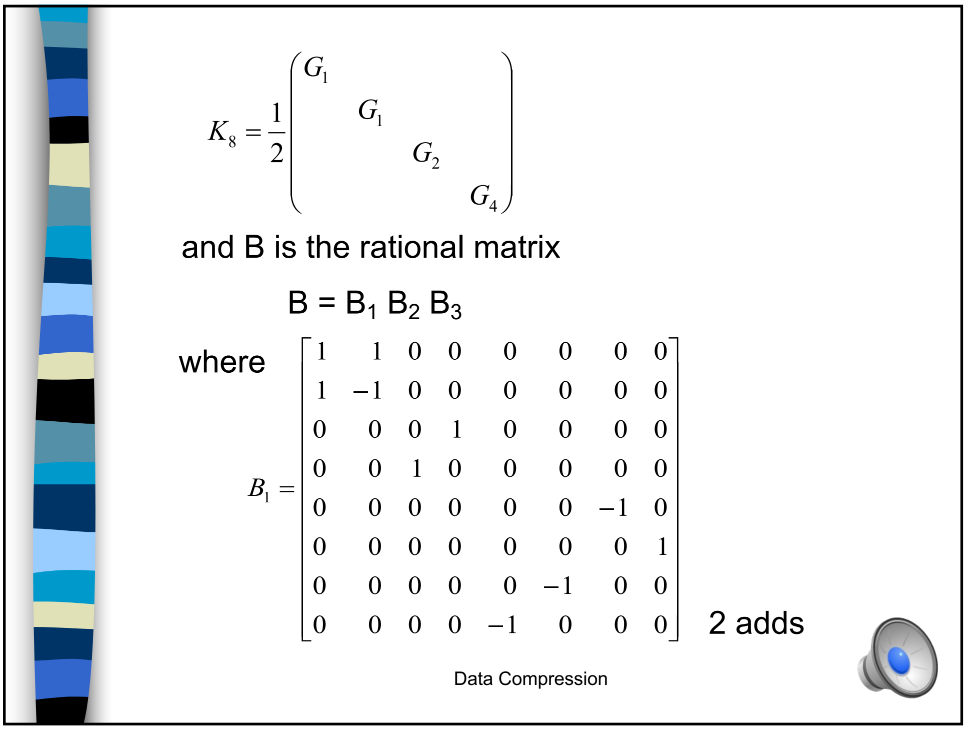
$$C_2F = 2\begin{pmatrix} r(4) & \\ & \boxed{r(4)} \end{pmatrix} \rightarrow \text{G(1)} \qquad (8)$$

Putting all these factorizations together yields, after slight arithmetic manipulation, the following factorization for the 8-point DCT matrix :

$$C_8 = P_8\, K_8\, B$$

where P_8 is the signed-permutation matrix

$$P_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



$$K_8 = \frac{1}{2}\begin{pmatrix} G_1 & & & \\ & G_1 & & \\ & & G_2 & \\ & & & G_4 \end{pmatrix}$$

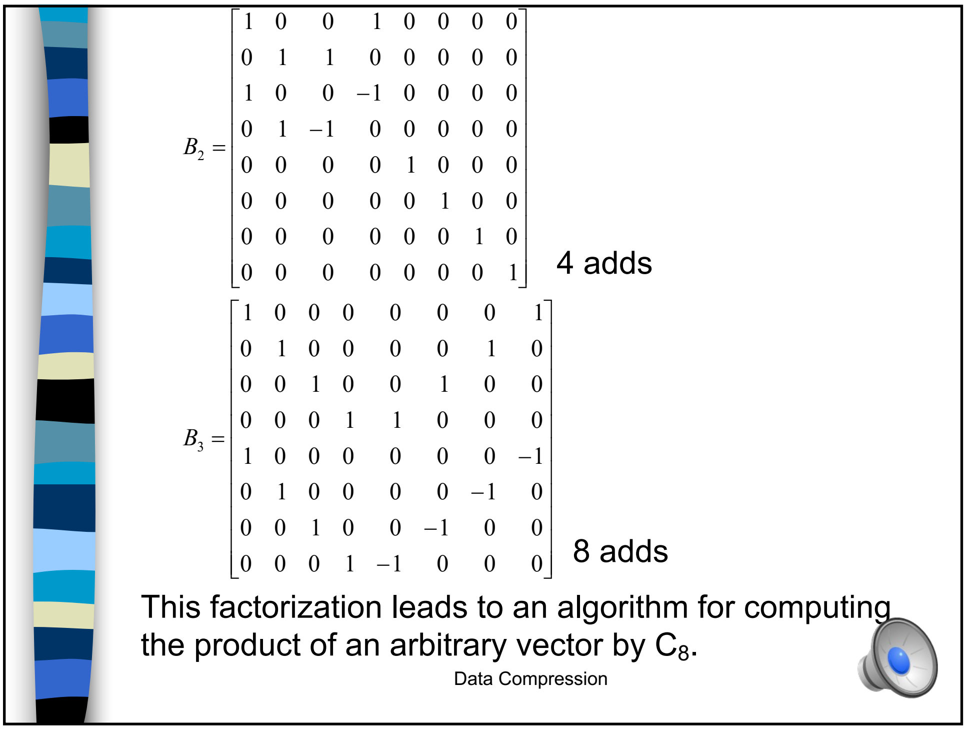
and B is the rational matrix

$$B = B_1\, B_2\, B_3$$

where

$$B_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

2 adds



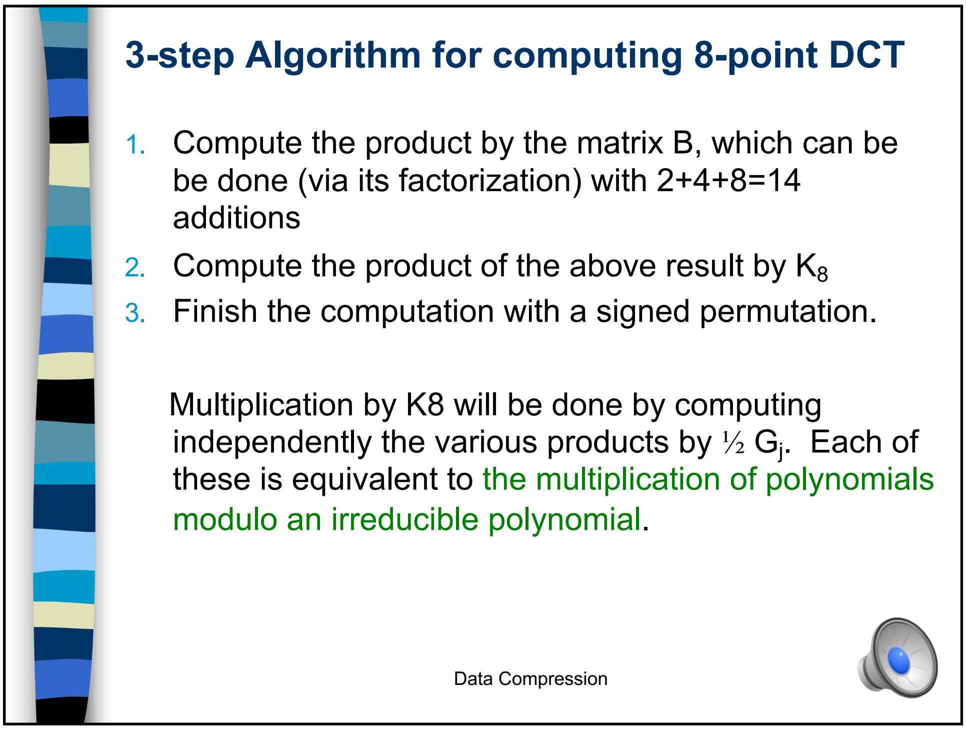
$$B_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4 adds

$$B_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

8 adds

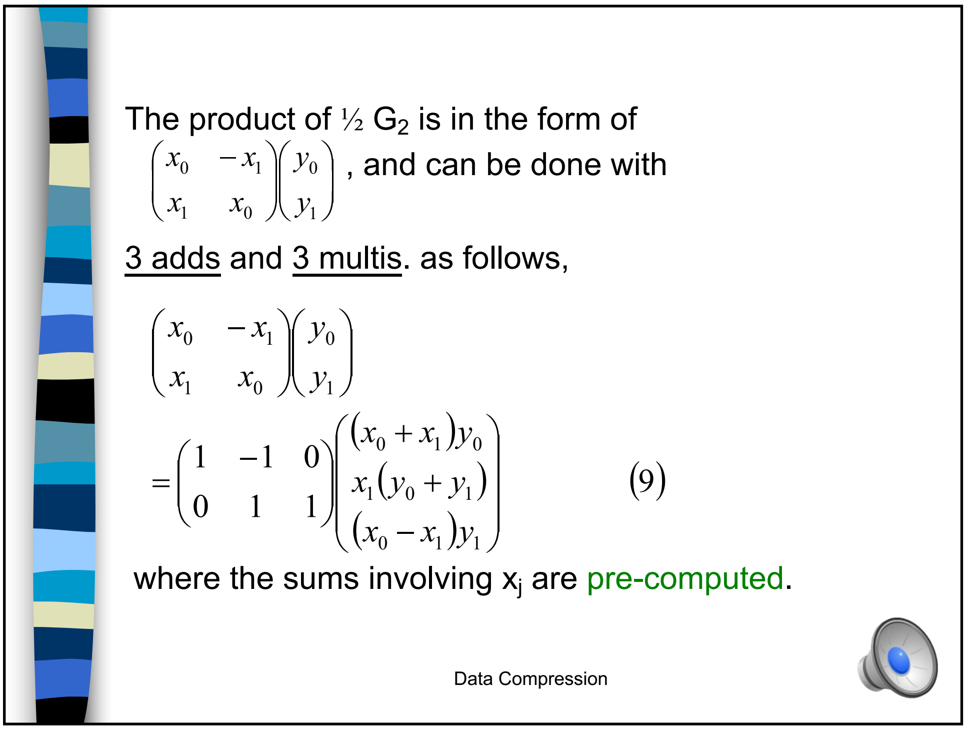
This factorization leads to an algorithm for computing the product of an arbitrary vector by C_8.



3-step Algorithm for computing 8-point DCT

1. Compute the product by the matrix B, which can be be done (via its factorization) with 2+4+8=14 additions
2. Compute the product of the above result by K_8
3. Finish the computation with a signed permutation.

Multiplication by K8 will be done by computing independently the various products by ½ G_j. Each of these is equivalent to the multiplication of polynomials modulo an irreducible polynomial.

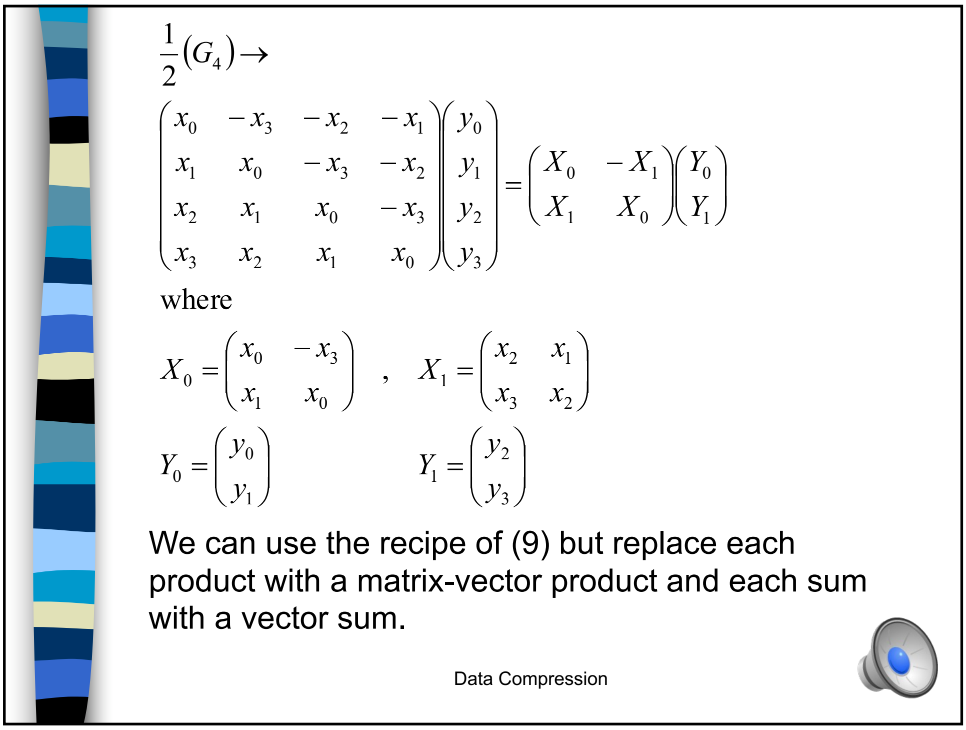


The product of ½ G$_2$ is in the form of $\begin{pmatrix} x_0 & -x_1 \\ x_1 & x_0 \end{pmatrix}\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}$, and can be done with

3 adds and 3 multis. as follows,

$$\begin{pmatrix} x_0 & -x_1 \\ x_1 & x_0 \end{pmatrix}\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} (x_0+x_1)y_0 \\ x_1(y_0+y_1) \\ (x_0-x_1)y_1 \end{pmatrix} \qquad (9)$$

where the sums involving x$_j$ are pre-computed.



$$\frac{1}{2}(G_4) \rightarrow$$

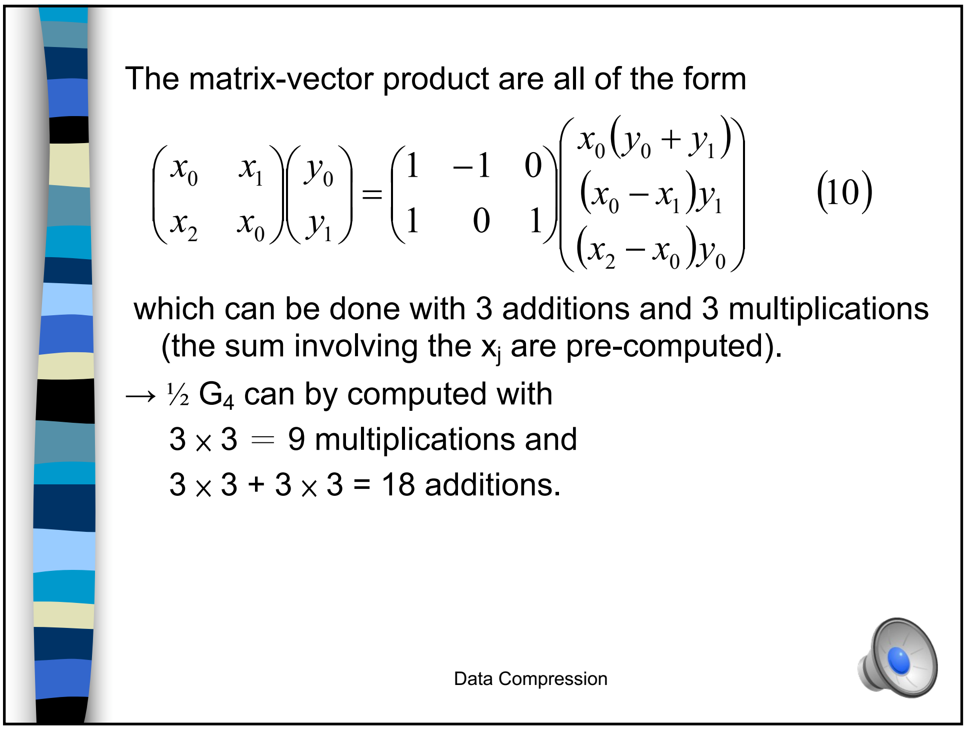
$$\begin{pmatrix} x_0 & -x_3 & -x_2 & -x_1 \\ x_1 & x_0 & -x_3 & -x_2 \\ x_2 & x_1 & x_0 & -x_3 \\ x_3 & x_2 & x_1 & x_0 \end{pmatrix}\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} X_0 & -X_1 \\ X_1 & X_0 \end{pmatrix}\begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}$$

where

$$X_0 = \begin{pmatrix} x_0 & -x_3 \\ x_1 & x_0 \end{pmatrix} \quad , \quad X_1 = \begin{pmatrix} x_2 & x_1 \\ x_3 & x_2 \end{pmatrix}$$

$$Y_0 = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \qquad Y_1 = \begin{pmatrix} y_2 \\ y_3 \end{pmatrix}$$

We can use the recipe of (9) but replace each product with a matrix-vector product and each sum with a vector sum.



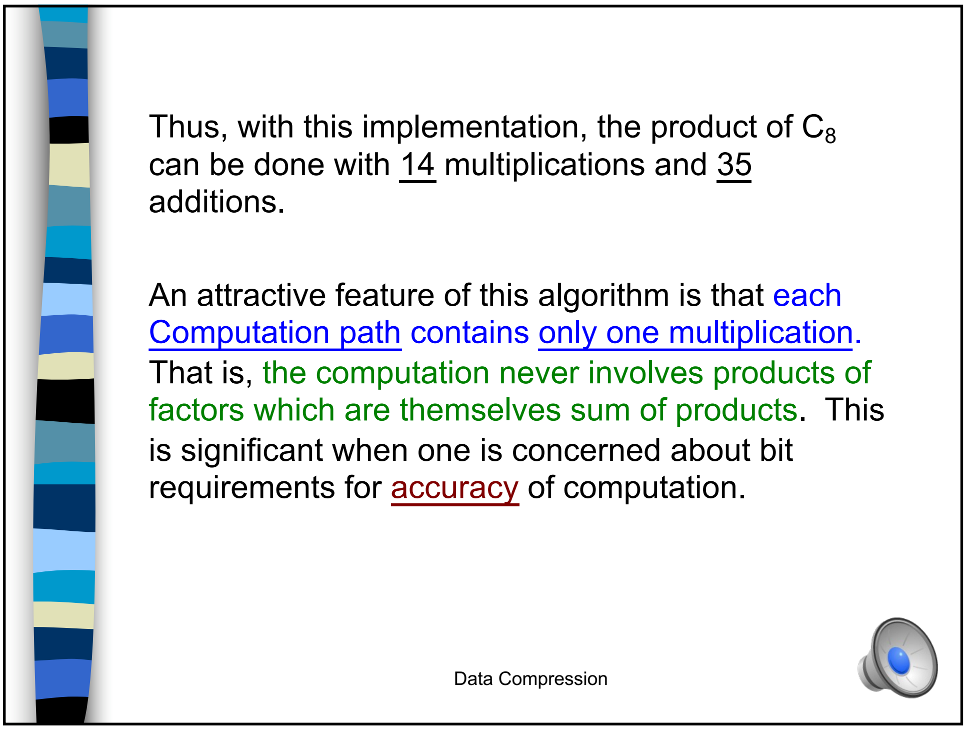
The matrix-vector product are all of the form

$$\begin{pmatrix} x_0 & x_1 \\ x_2 & x_0 \end{pmatrix}\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_0(y_0+y_1) \\ (x_0-x_1)y_1 \\ (x_2-x_0)y_0 \end{pmatrix} \qquad (10)$$

which can be done with 3 additions and 3 multiplications (the sum involving the x_j are pre-computed).

→ ½ G_4 can by computed with
$3 \times 3 = 9$ multiplications and
$3 \times 3 + 3 \times 3 = 18$ additions.

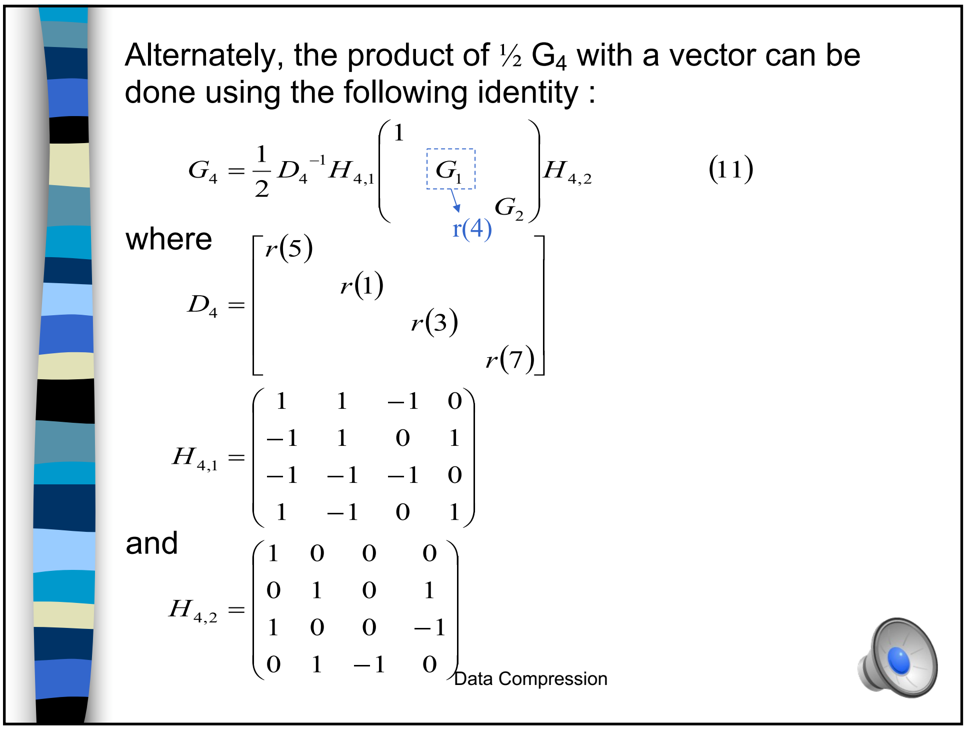
19



Thus, with this implementation, the product of C_8 can be done with 14 multiplications and 35 additions.

An attractive feature of this algorithm is that each Computation path contains only one multiplication. That is, the computation never involves products of factors which are themselves sum of products. This is significant when one is concerned about bit requirements for accuracy of computation.

20



Alternately, the product of ½ G_4 with a vector can be done using the following identity :

$$G_4 = \frac{1}{2} D_4^{-1} H_{4,1} \begin{pmatrix} 1 & & \\ & \underset{\downarrow\, r(4)}{\boxed{G_1}} & \\ & & G_2 \end{pmatrix} H_{4,2} \qquad (11)$$

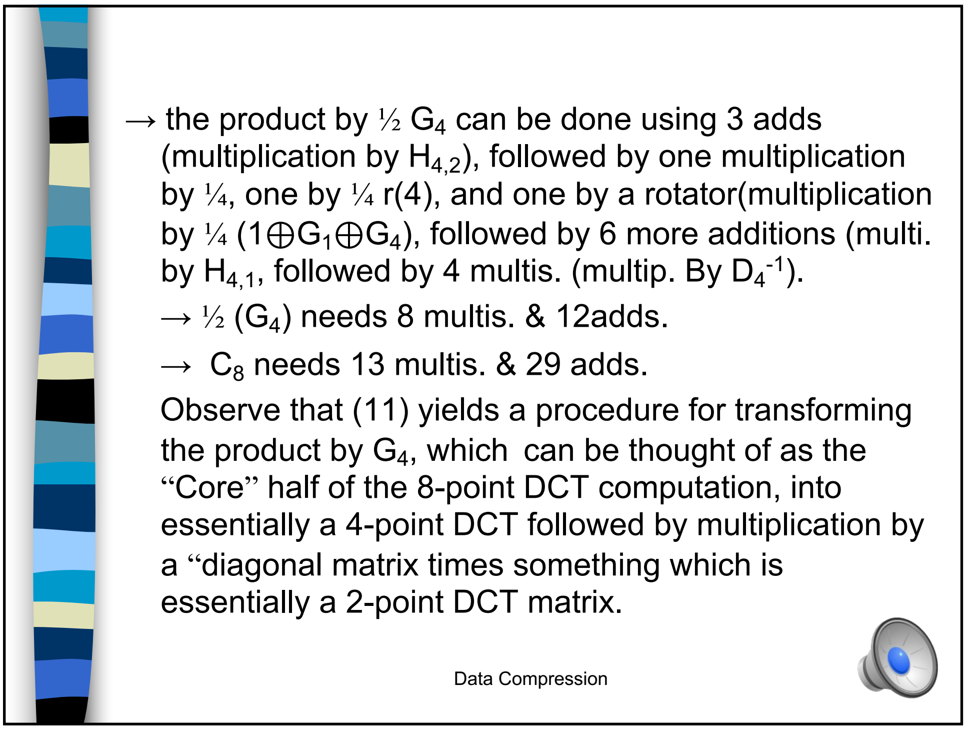
where

$$D_4 = \begin{bmatrix} r(5) & & & \\ & r(1) & & \\ & & r(3) & \\ & & & r(7) \end{bmatrix}$$

$$H_{4,1} = \begin{pmatrix} 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}$$

and

$$H_{4,2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

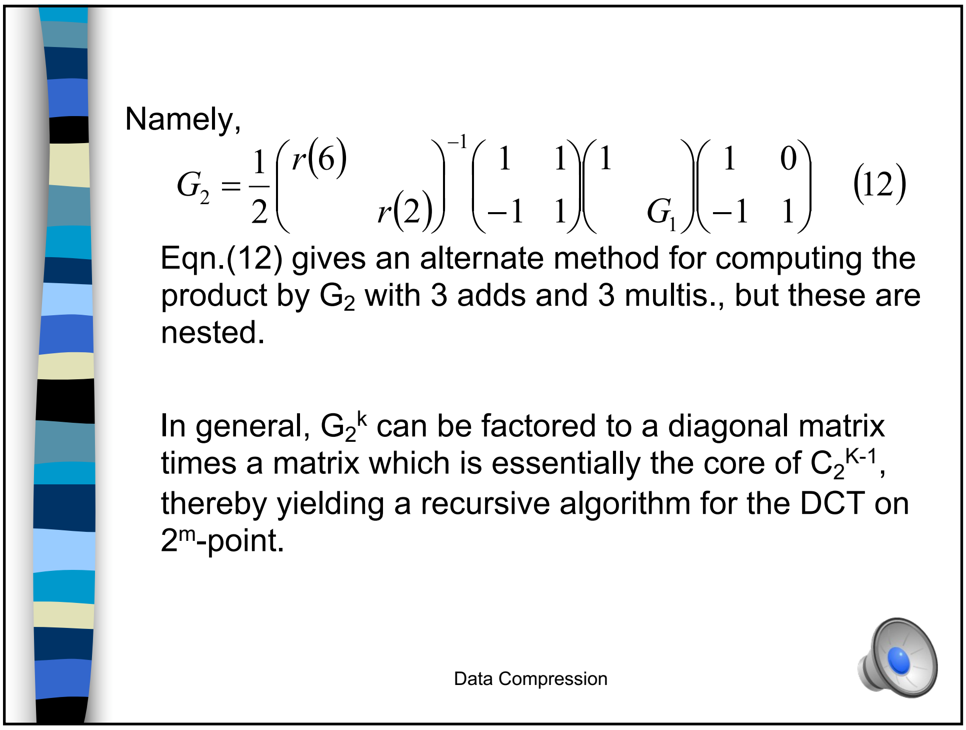


→ the product by ½ G_4 can be done using 3 adds (multiplication by $H_{4,2}$), followed by one multiplication by ¼, one by ¼ r(4), and one by a rotator(multiplication by ¼ $(1 \oplus G_1 \oplus G_4)$, followed by 6 more additions (multi. by $H_{4,1}$, followed by 4 multis. (multip. By D_4^{-1}).

→ ½ (G_4) needs 8 multis. & 12adds.

→ C_8 needs 13 multis. & 29 adds.

Observe that (11) yields a procedure for transforming the product by G_4, which can be thought of as the "Core" half of the 8-point DCT computation, into essentially a 4-point DCT followed by multiplication by a "diagonal matrix times something which is essentially a 2-point DCT matrix.

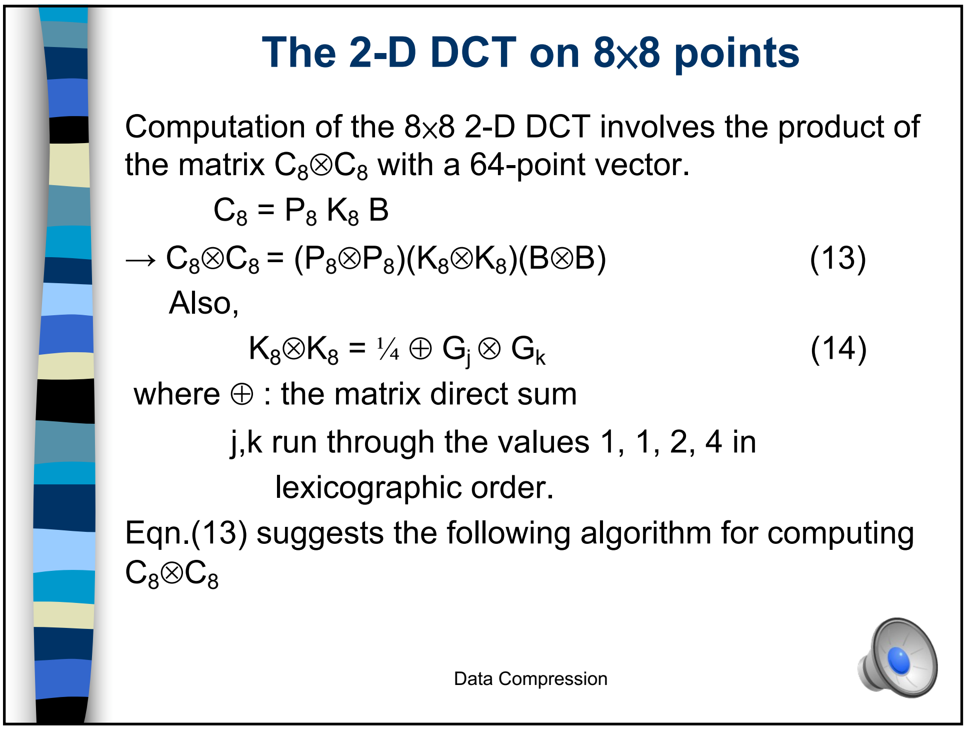


Namely,

$$G_2 = \frac{1}{2}\begin{pmatrix} r(6) & \\ & r(2) \end{pmatrix}^{-1}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & \\ & G_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \quad (12)$$

Eqn.(12) gives an alternate method for computing the product by G_2 with 3 adds and 3 multis., but these are nested.

In general, G_{2^k} can be factored to a diagonal matrix times a matrix which is essentially the core of $C_{2^{K-1}}$, thereby yielding a recursive algorithm for the DCT on 2^m-point.



The 2-D DCT on 8×8 points

Computation of the 8×8 2-D DCT involves the product of the matrix $C_8 \otimes C_8$ with a 64-point vector.

$$C_8 = P_8 K_8 B$$

$$\rightarrow C_8 \otimes C_8 = (P_8 \otimes P_8)(K_8 \otimes K_8)(B \otimes B) \quad (13)$$

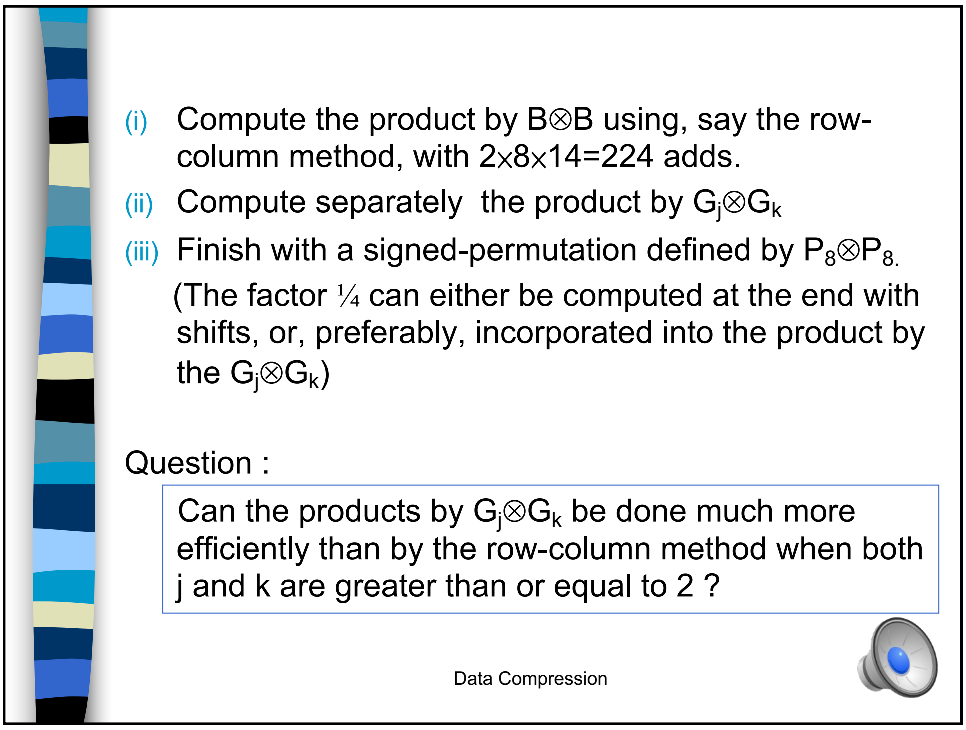
Also,

$$K_8 \otimes K_8 = \tfrac{1}{4} \oplus G_j \otimes G_k \quad (14)$$

where ⊕ : the matrix direct sum

j,k run through the values 1, 1, 2, 4 in lexicographic order.

Eqn.(13) suggests the following algorithm for computing $C_8 \otimes C_8$



(i) Compute the product by $B\otimes B$ using, say the row-column method, with 2×8×14=224 adds.

(ii) Compute separately the product by $G_j\otimes G_k$

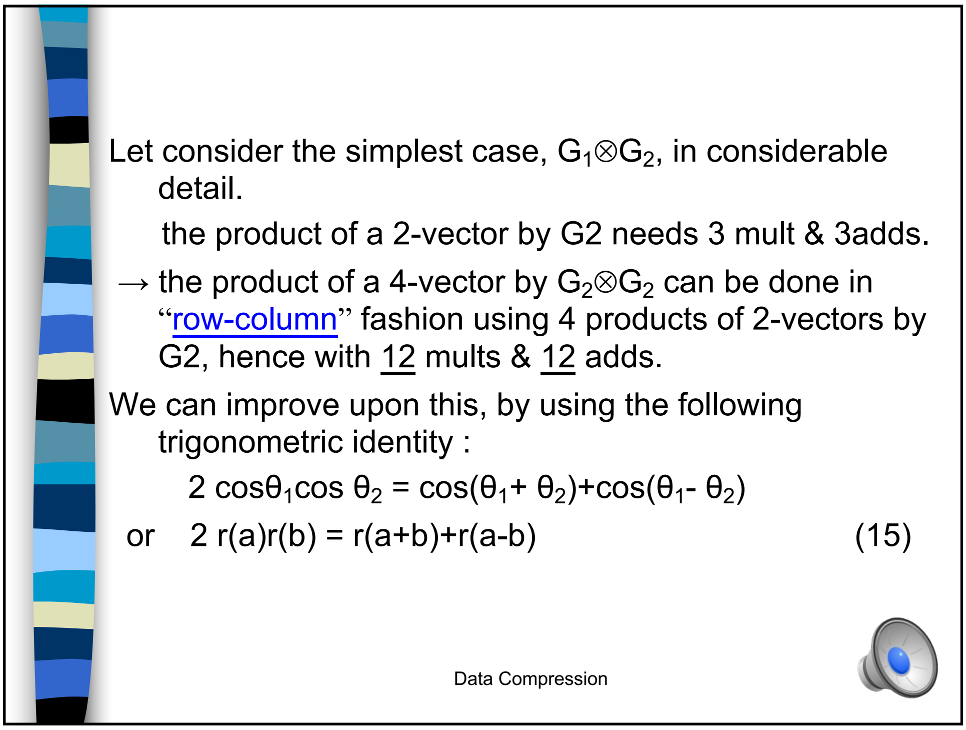
(iii) Finish with a signed-permutation defined by $P_8\otimes P_8$.

(The factor ¼ can either be computed at the end with shifts, or, preferably, incorporated into the product by the $G_j\otimes G_k$)

Question :

Can the products by $G_j\otimes G_k$ be done much more efficiently than by the row-column method when both j and k are greater than or equal to 2 ?

25



Let consider the simplest case, $G_1\otimes G_2$, in considerable detail.

the product of a 2-vector by G2 needs 3 mult & 3adds.

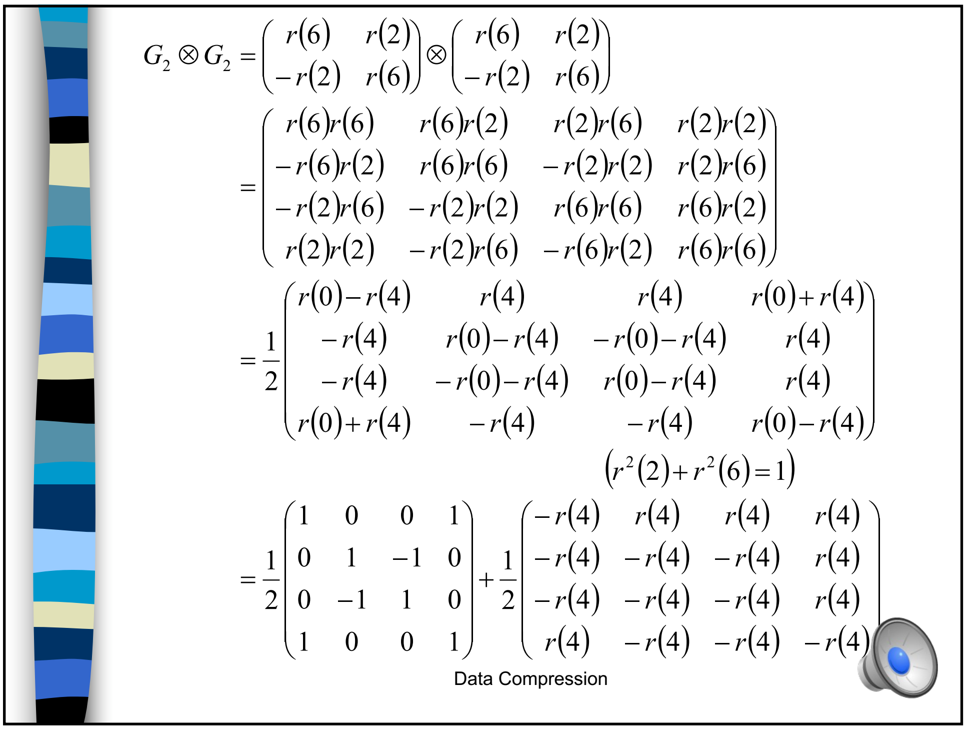
→ the product of a 4-vector by $G_2\otimes G_2$ can be done in "row-column" fashion using 4 products of 2-vectors by G2, hence with <u>12</u> mults & <u>12</u> adds.

We can improve upon this, by using the following trigonometric identity :

$$2 \cos\theta_1 \cos \theta_2 = \cos(\theta_1+ \theta_2)+\cos(\theta_1- \theta_2)$$

or $$2\ r(a)r(b) = r(a+b)+r(a-b) \qquad (15)$$

26



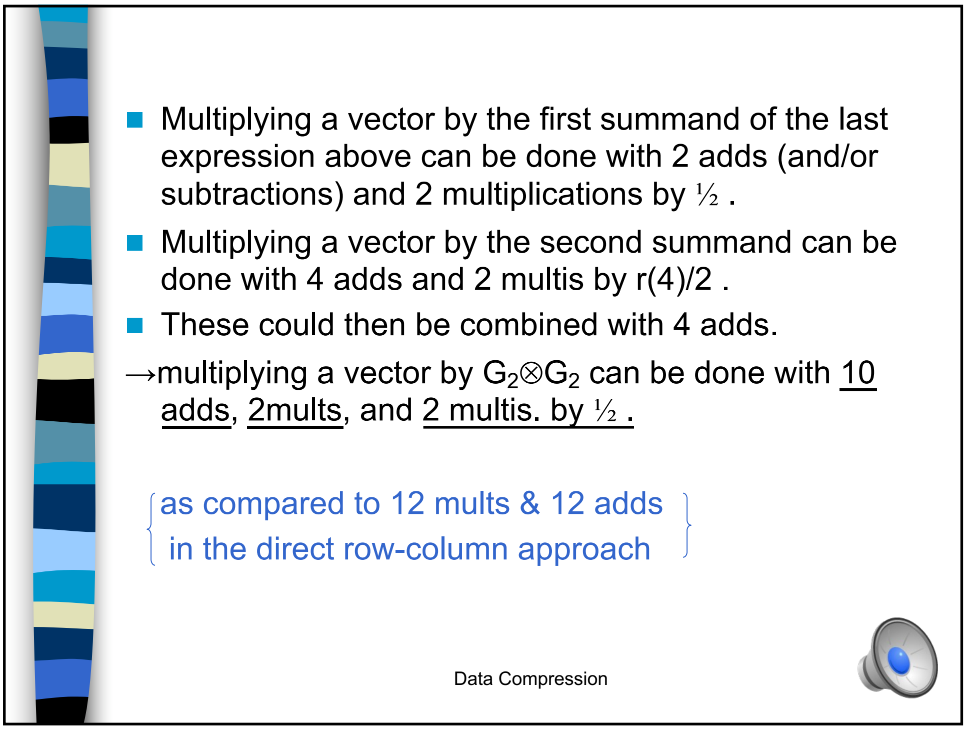
$$G_2 \otimes G_2 = \begin{pmatrix} r(6) & r(2) \\ -r(2) & r(6) \end{pmatrix} \otimes \begin{pmatrix} r(6) & r(2) \\ -r(2) & r(6) \end{pmatrix}$$

$$= \begin{pmatrix} r(6)r(6) & r(6)r(2) & r(2)r(6) & r(2)r(2) \\ -r(6)r(2) & r(6)r(6) & -r(2)r(2) & r(2)r(6) \\ -r(2)r(6) & -r(2)r(2) & r(6)r(6) & r(6)r(2) \\ r(2)r(2) & -r(2)r(6) & -r(6)r(2) & r(6)r(6) \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} r(0)-r(4) & r(4) & r(4) & r(0)+r(4) \\ -r(4) & r(0)-r(4) & -r(0)-r(4) & r(4) \\ -r(4) & -r(0)-r(4) & r(0)-r(4) & r(4) \\ r(0)+r(4) & -r(4) & -r(4) & r(0)-r(4) \end{pmatrix}$$

$$\left(r^2(2)+r^2(6)=1\right)$$

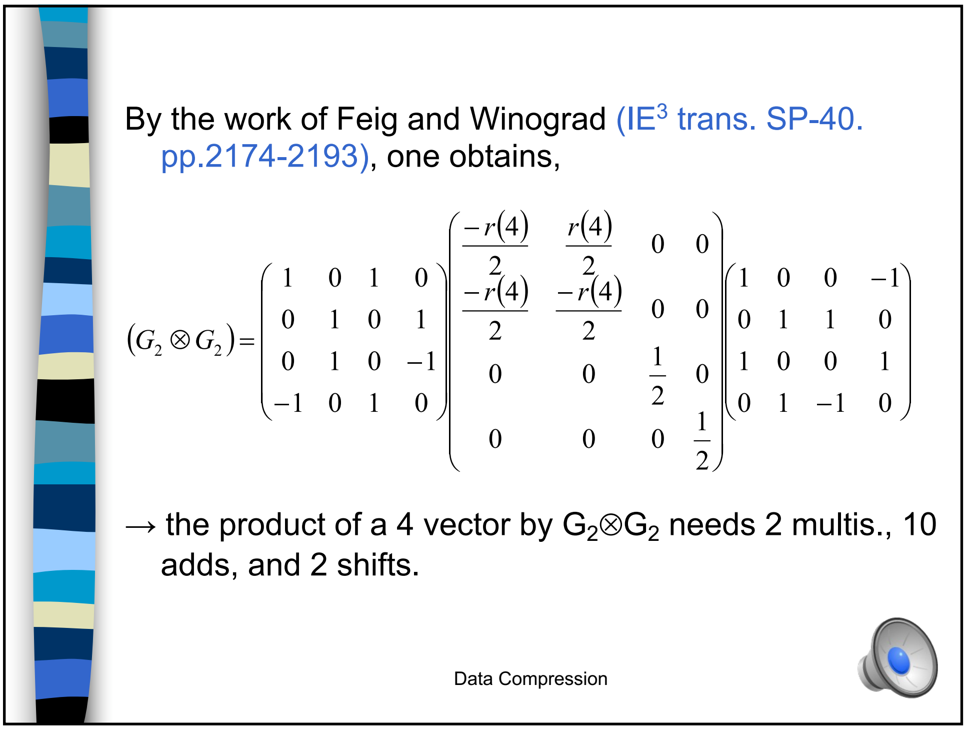
$$= \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} -r(4) & r(4) & r(4) & r(4) \\ -r(4) & -r(4) & -r(4) & r(4) \\ -r(4) & -r(4) & -r(4) & r(4) \\ r(4) & -r(4) & -r(4) & -r(4) \end{pmatrix}$$



- Multiplying a vector by the first summand of the last expression above can be done with 2 adds (and/or subtractions) and 2 multiplications by ½ .
- Multiplying a vector by the second summand can be done with 4 adds and 2 multis by r(4)/2 .
- These could then be combined with 4 adds.

→multiplying a vector by $G_2 \otimes G_2$ can be done with 10 adds, 2mults, and 2 multis. by ½ .

{ as compared to 12 mults & 12 adds in the direct row-column approach }



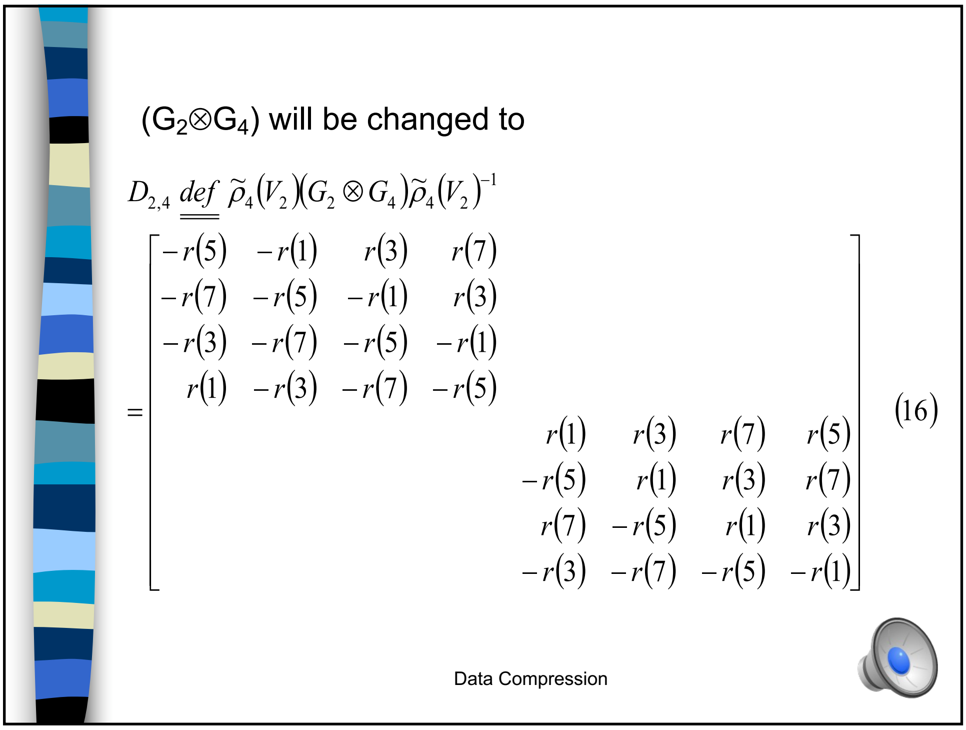
By the work of Feig and Winograd (IE^3 trans. SP-40. pp.2174-2193), one obtains,

$$(G_2 \otimes G_2) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{-r(4)}{2} & \frac{r(4)}{2} & 0 & 0 \\ \frac{-r(4)}{2} & \frac{-r(4)}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

→ the product of a 4 vector by $G_2 \otimes G_2$ needs 2 multis., 10 adds, and 2 shifts.

Data Compression

29



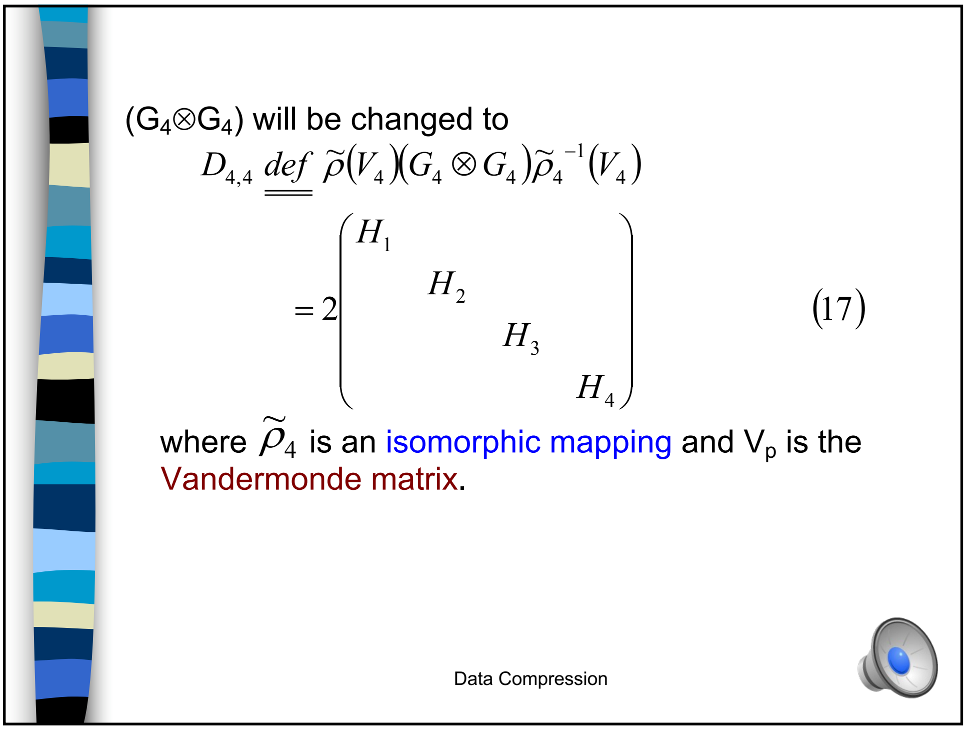
$(G_2 \otimes G_4)$ will be changed to

$$D_{2,4} \underset{=}{\underline{def}} \tilde{\rho}_4(V_2)(G_2 \otimes G_4)\tilde{\rho}_4(V_2)^{-1}$$

$$= \begin{bmatrix} -r(5) & -r(1) & r(3) & r(7) & & & & \\ -r(7) & -r(5) & -r(1) & r(3) & & & & \\ -r(3) & -r(7) & -r(5) & -r(1) & & & & \\ r(1) & -r(3) & -r(7) & -r(5) & & & & \\ & & & & r(1) & r(3) & r(7) & r(5) \\ & & & & -r(5) & r(1) & r(3) & r(7) \\ & & & & r(7) & -r(5) & r(1) & r(3) \\ & & & & -r(3) & -r(7) & -r(5) & -r(1) \end{bmatrix} \quad (16)$$

Data Compression

30

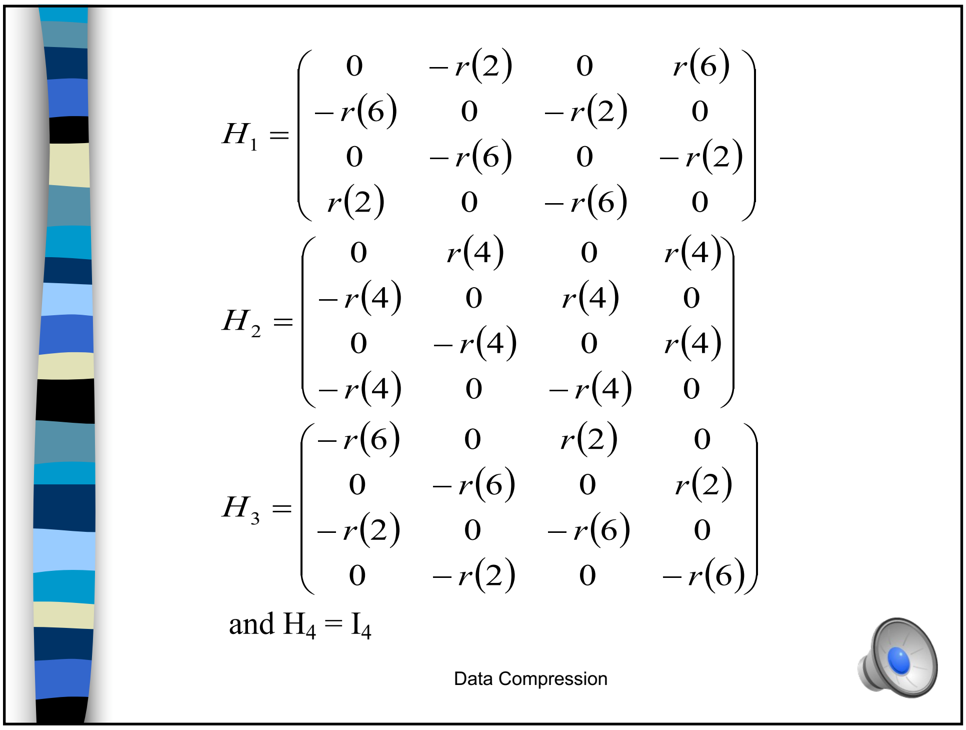


($G_4 \otimes G_4$) will be changed to

$$D_{4,4} \underset{=}{def} \tilde{\rho}(V_4)(G_4 \otimes G_4)\tilde{\rho}_4^{-1}(V_4)$$

$$= 2\begin{pmatrix} H_1 & & & \\ & H_2 & & \\ & & H_3 & \\ & & & H_4 \end{pmatrix} \qquad (17)$$

where $\tilde{\rho}_4$ is an isomorphic mapping and V_p is the Vandermonde matrix.



$$H_1 = \begin{pmatrix} 0 & -r(2) & 0 & r(6) \\ -r(6) & 0 & -r(2) & 0 \\ 0 & -r(6) & 0 & -r(2) \\ r(2) & 0 & -r(6) & 0 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} 0 & r(4) & 0 & r(4) \\ -r(4) & 0 & r(4) & 0 \\ 0 & -r(4) & 0 & r(4) \\ -r(4) & 0 & -r(4) & 0 \end{pmatrix}$$

$$H_3 = \begin{pmatrix} -r(6) & 0 & r(2) & 0 \\ 0 & -r(6) & 0 & r(2) \\ -r(2) & 0 & -r(6) & 0 \\ 0 & -r(2) & 0 & -r(6) \end{pmatrix}$$

and $H_4 = I_4$

(16) → the product by $G_2 \otimes G_4$ is equivalent to 2 products by G_4

(17) → the product by $G_4 \otimes G_4$ is equivalent to a direct sum of 4 products by G_2 and 4 products by r(4).

$G_4 \otimes G_2 = P_s(G_2 \otimes G_4) P_s^{-1}$

algorithmically equivalent

where P_s : perfect-shuffle permutation

$$G_2 \otimes G_2 = \left(2\tilde{\rho}_2(v_2)\right)^{-1}\left(\frac{1}{2}D_{2,2}\right)\tilde{\rho}_2(v_2)$$

$$G_2 \otimes G_4 = \left(2\tilde{\rho}_4(v_2)\right)^{-1}\left(\frac{1}{2}D_{2,4}\right)\tilde{\rho}_4(v_2)$$

$$G_4 \otimes G_4 = \left(4\tilde{\rho}_4(v_4)\right)^{-1}\left(\frac{1}{4}D_{4,4}\right)\tilde{\rho}_4(v_4)$$

where

$$v_2 = \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$v_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & w^3 \end{pmatrix}$$

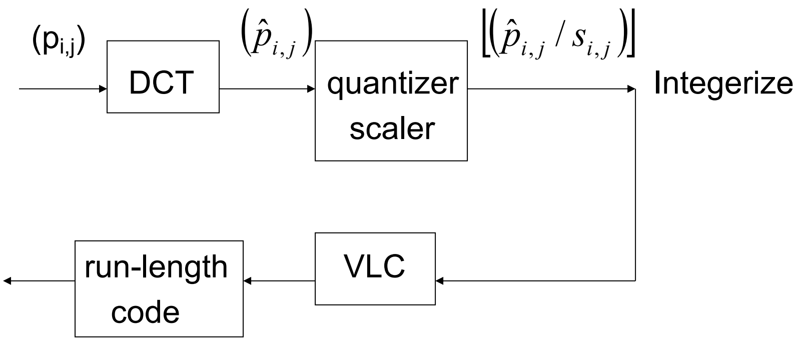
$$w = \exp\left(-j\frac{2\pi}{8}\right)$$

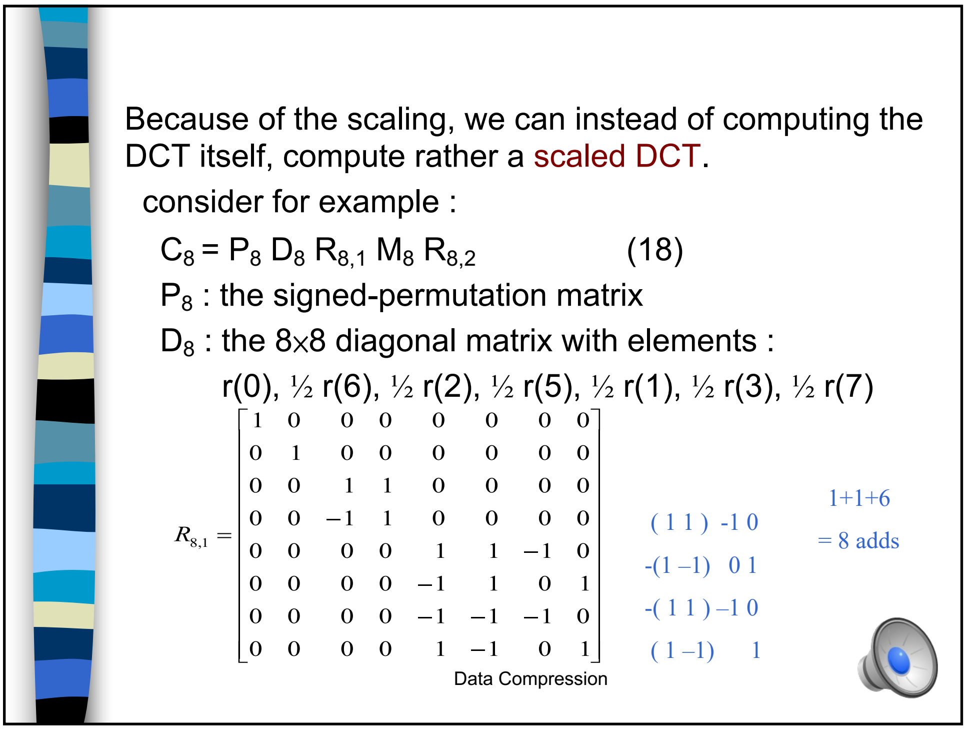
Altogether, the algorithm just described for the 8x8-point DCT will require 94 multiplications and 454 additions.

No. Times	Operator	Mults	Adds	Shifts	Total Mults	Total adds	Total shifts
4	$G_1 \otimes G_1$	0	0	1	0	0	4
4	$\sqrt{2}G_2$	3	3	0	12	12	0
4	$\sqrt{2}G_4$	8	12	0	32	48	0
1	$G_2 \otimes G_2$	2	10	2	2	10	2
2	$G_2 \otimes G_4$	16	40	0	32	80	0
1	$G_4 \otimes G_4$	16	80	0	16	80	0
	Pre-additions					224	
total					94	454	6

The Scaled DCT

In most applications, the DCT is followed by scaling and quantization.





Because of the scaling, we can instead of computing the DCT itself, compute rather a scaled DCT.

consider for example :

$C_8 = P_8 D_8 R_{8,1} M_8 R_{8,2}$ (18)

P_8 : the signed-permutation matrix

D_8 : the 8×8 diagonal matrix with elements :

r(0), ½ r(6), ½ r(2), ½ r(5), ½ r(1), ½ r(3), ½ r(7)

$$R_{8,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \end{bmatrix}$$

(1 1) -1 0

-(1 –1) 0 1

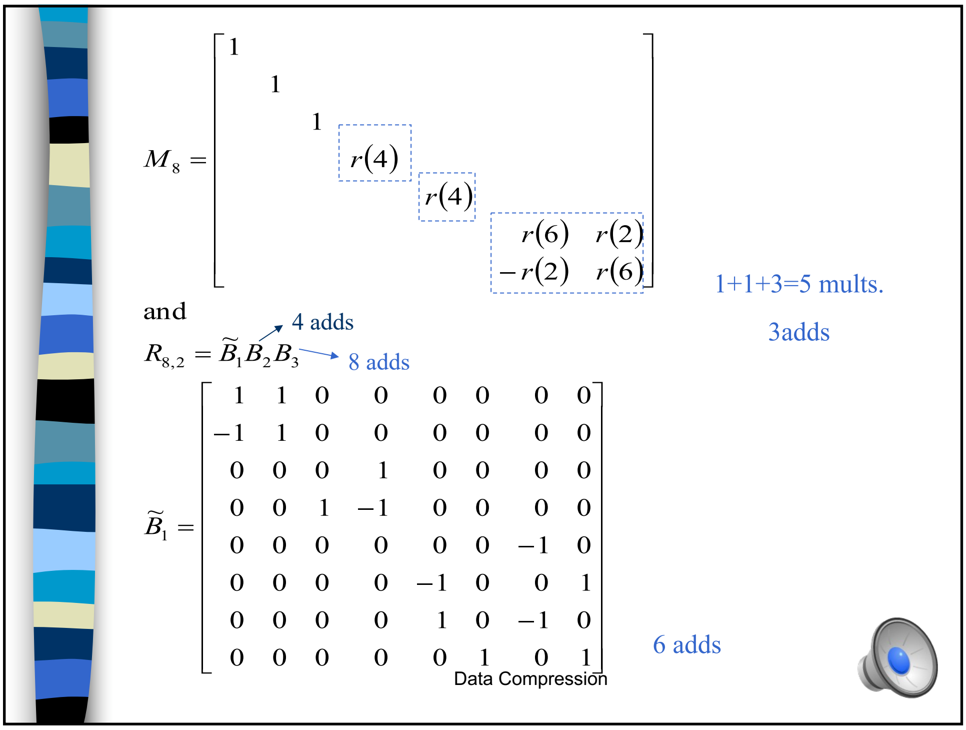
-(1 1) –1 0

(1 –1) 1

1+1+6

= 8 adds

37



$$M_8 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & \boxed{r(4)} & & & & \\ & & & & \boxed{r(4)} & & & \\ & & & & & & r(6) & r(2) \\ & & & & & & -r(2) & r(6) \end{bmatrix}$$

1+1+3=5 mults.

3adds

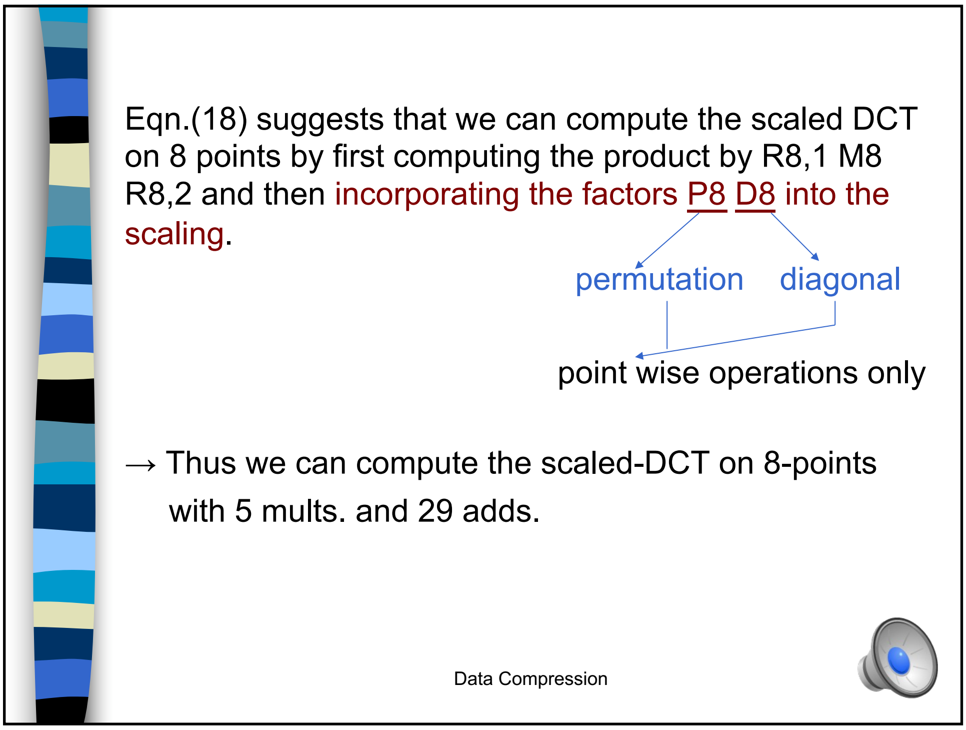
and

$$R_{8,2} = \widetilde{B}_1 \overset{\text{4 adds}}{B_2} B_3 \rightarrow \text{8 adds}$$

$$\widetilde{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

6 adds

38

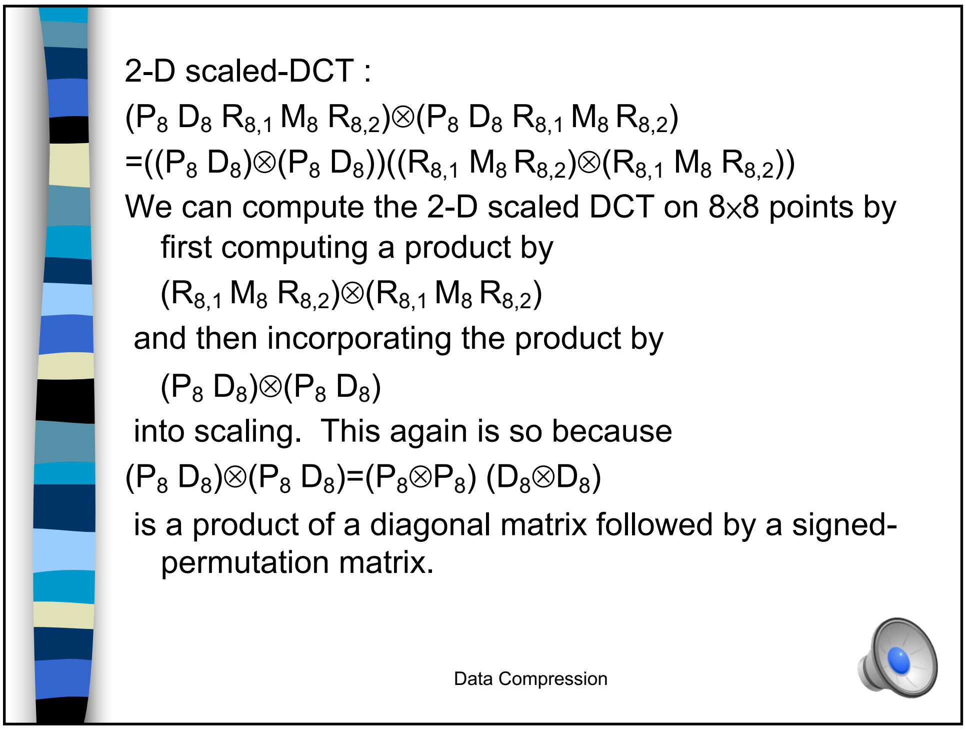


Eqn.(18) suggests that we can compute the scaled DCT on 8 points by first computing the product by R8,1 M8 R8,2 and then incorporating the factors P8 D8 into the scaling.

permutation diagonal

point wise operations only

→ Thus we can compute the scaled-DCT on 8-points with 5 mults. and 29 adds.



2-D scaled-DCT :

$(P_8\ D_8\ R_{8,1}\ M_8\ R_{8,2})\otimes(P_8\ D_8\ R_{8,1}\ M_8\ R_{8,2})$

$=((P_8\ D_8)\otimes(P_8\ D_8))((R_{8,1}\ M_8\ R_{8,2})\otimes(R_{8,1}\ M_8\ R_{8,2}))$

We can compute the 2-D scaled DCT on 8x8 points by first computing a product by

$(R_{8,1}\ M_8\ R_{8,2})\otimes(R_{8,1}\ M_8\ R_{8,2})$

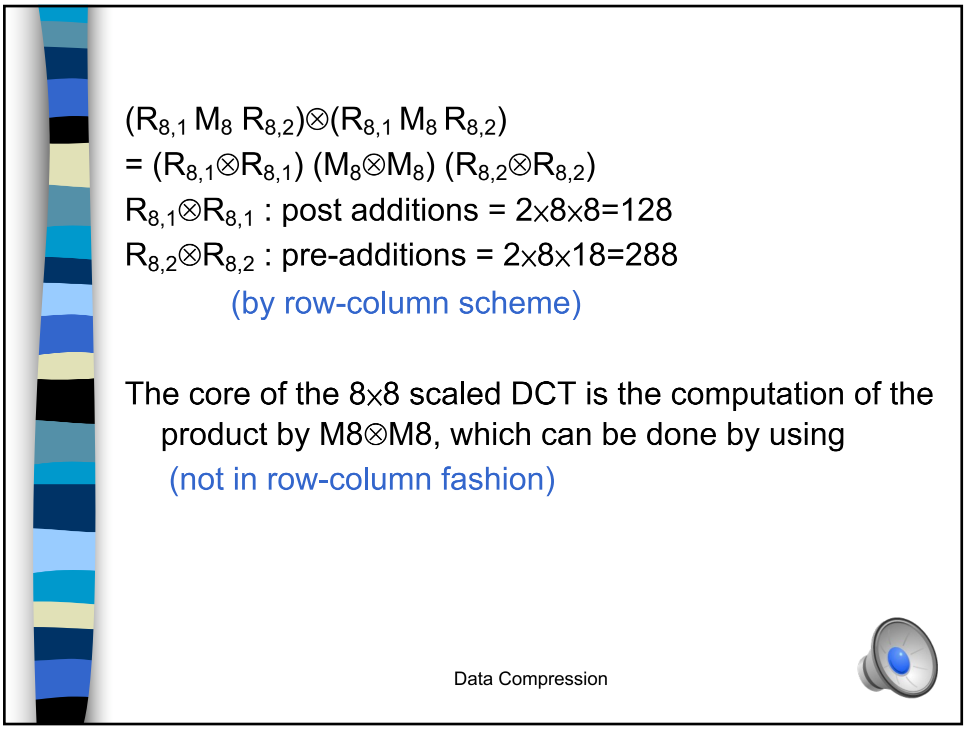
and then incorporating the product by

$(P_8\ D_8)\otimes(P_8\ D_8)$

into scaling. This again is so because

$(P_8\ D_8)\otimes(P_8\ D_8)=(P_8\otimes P_8)\ (D_8\otimes D_8)$

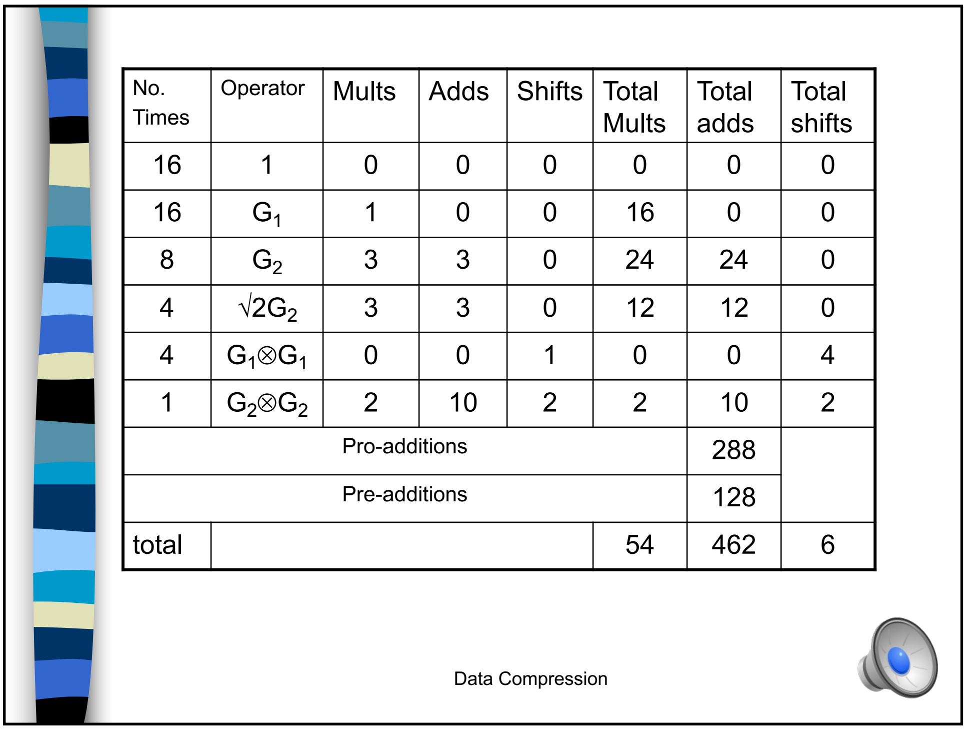
is a product of a diagonal matrix followed by a signed-permutation matrix.



$(R_{8,1} M_8 R_{8,2}) \otimes (R_{8,1} M_8 R_{8,2})$
$= (R_{8,1} \otimes R_{8,1})\ (M_8 \otimes M_8)\ (R_{8,2} \otimes R_{8,2})$
$R_{8,1} \otimes R_{8,1}$: post additions = 2x8x8=128
$R_{8,2} \otimes R_{8,2}$: pre-additions = 2x8x18=288
(by row-column scheme)

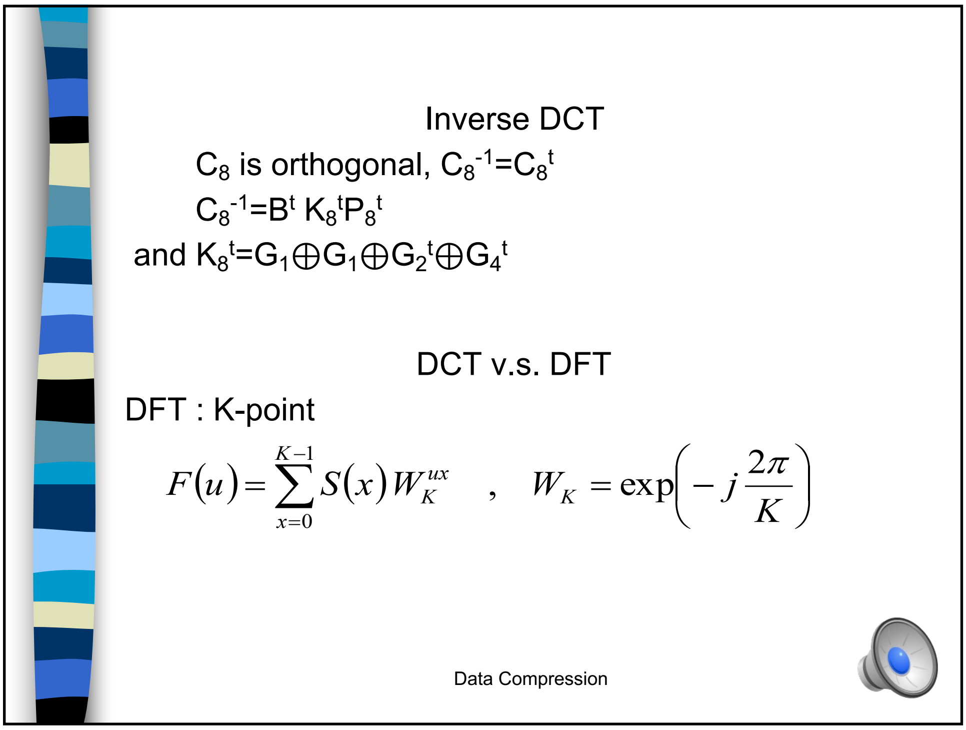
The core of the 8x8 scaled DCT is the computation of the product by M8⊗M8, which can be done by using (not in row-column fashion)

41



No. Times	Operator	Mults	Adds	Shifts	Total Mults	Total adds	Total shifts
16	1	0	0	0	0	0	0
16	G_1	1	0	0	16	0	0
8	G_2	3	3	0	24	24	0
4	$\sqrt{}2G_2$	3	3	0	12	12	0
4	$G_1 \otimes G_1$	0	0	1	0	0	4
1	$G_2 \otimes G_2$	2	10	2	2	10	2
Pro-additions						288	
Pre-additions						128	
total					54	462	6

42



Inverse DCT

C_8 is orthogonal, $C_8^{-1}=C_8^t$

$C_8^{-1}=B^t K_8^t P_8^t$

and $K_8^t=G_1 \oplus G_1 \oplus G_2^t \oplus G_4^t$

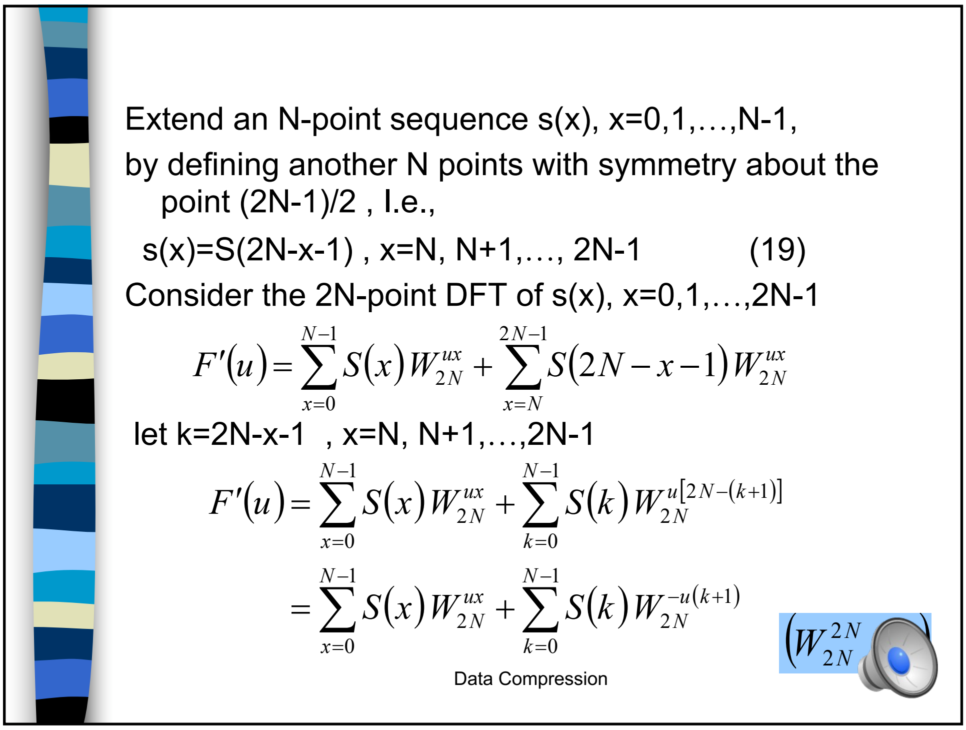
DCT v.s. DFT

DFT : K-point

$$F(u)=\sum_{x=0}^{K-1} S(x)\, W_K^{ux} \quad , \quad W_K = \exp\left(-j\frac{2\pi}{K}\right)$$

Data Compression

43



Extend an N-point sequence s(x), x=0,1,…,N-1,

by defining another N points with symmetry about the point (2N-1)/2 , I.e.,

s(x)=S(2N-x-1) , x=N, N+1,…, 2N-1 (19)

Consider the 2N-point DFT of s(x), x=0,1,…,2N-1

$$F'(u)=\sum_{x=0}^{N-1} S(x)\, W_{2N}^{ux} + \sum_{x=N}^{2N-1} S(2N-x-1)\, W_{2N}^{ux}$$

let k=2N-x-1 , x=N, N+1,…,2N-1

$$F'(u)=\sum_{x=0}^{N-1} S(x)\, W_{2N}^{ux} + \sum_{k=0}^{N-1} S(k)\, W_{2N}^{u[2N-(k+1)]}$$

$$=\sum_{x=0}^{N-1} S(x)\, W_{2N}^{ux} + \sum_{k=0}^{N-1} S(k)\, W_{2N}^{-u(k+1)}$$

(W_{2N}^{2N})

Data Compression

44

replacing the index k by x and multiplying by $\frac{1}{2}W_{2N}^{\frac{u}{2}}$

$$\rightarrow \frac{1}{2}F'(u)W_{2N}^{\frac{u}{2}} = \sum_{x=0}^{N-1} S(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \qquad (20)$$

→ The first N DFT coeffs., when multiplied by a complex scaling factor, give the N-point DCT coeffs.

Notice that, the right-hand side of Eqn.(20) is real, the left-hand side must also be real.

Denoting the real and imaginary part F(u) by A(u) and B(u) respectively,

$$F(u)W_{2N}^{\frac{u}{2}} = (A_u + jB_u)\cdot\left(\cos\left(\pi u/_{2N}\right) - j\sin\left(\frac{\pi u}{2N}\right)\right) \qquad (21)$$

Expanding the above equation into separate real and imaginary parts and setting the imaginary part to zero (the DCT is real),

we get :

$$B_u = A_u \, \sin\left(\pi u/_{2N}\right) \Big/ \cos\left(\pi u/_{2N}\right)$$

When this is substituted back into eqn.(21), it gives :

$$F(u)W_{2N}^{\frac{u}{2}} = A_u \sec\left(\pi u/_{2N}\right)$$

$$= R_e(F(u))\sec\left(\frac{\pi u}{2N}\right)$$

$$\Rightarrow \sum_{x=0}^{N-1} S(x)\cos\left[\frac{\pi u(2x+1)}{2N}\right]$$

$$= R_e[F(u)]\cdot\sec\left(\frac{\pi u}{2N}\right)$$

→ The DCT coeffs. can be obtained by a simple

N-point

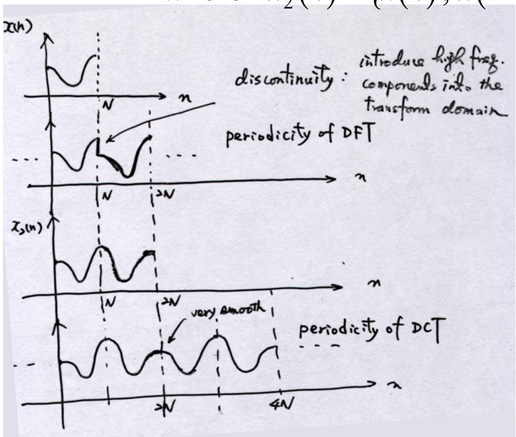
scaling of the real part of the DFT coeffs.

2N-point

The relationship between DFT and DCT, i.e. eqn.(20), explains why DCTs outperform DFTs in terms of energy compaction :

$$DCT_N\{x(n)\} = W_{2N}^{\frac{k}{2}}\ DFT_{2N}\{x_2(n)\}$$

$$\text{where}\quad x_2(n) = \{x(n),\, x(2N-1-n)\}$$


x(n)
n
N
discontinuity : introduce high freq. components into the transform domain
periodicity of DFT
n
N
2N
x2(n)
n
N
2N
very smooth
periodicity of DCT
n
2N
4N