北京大学数字媒体研究所
INSTITUTE OF DIGITAL MEDIA, PEKING UNIVERSITY

Simplified multiplier-less 4x4 DST for intra prediction residue

Xin Zhao, Li Zhang, Siwei Ma, Wen Gao

Institute of Digital Media, Peking University

Jan 2011

Outline

- Algorithm
- Simulation results
- Complexity
- Summary

Algorithm

- Mode-dependent transform

$$\boldsymbol{F}=\boldsymbol{D}\cdot\boldsymbol{X}\cdot\boldsymbol{D}^{\mathrm{T}}$$

	Transform matrix ***D***	
	DC prediction	Others
4x4 modes (0~17)	DCT (TMuC 0.9)	S_4
8x8 modes (0~33)	DCT (TMuC 0.9)	S_8

Algorithm

- Mode-dependent transform

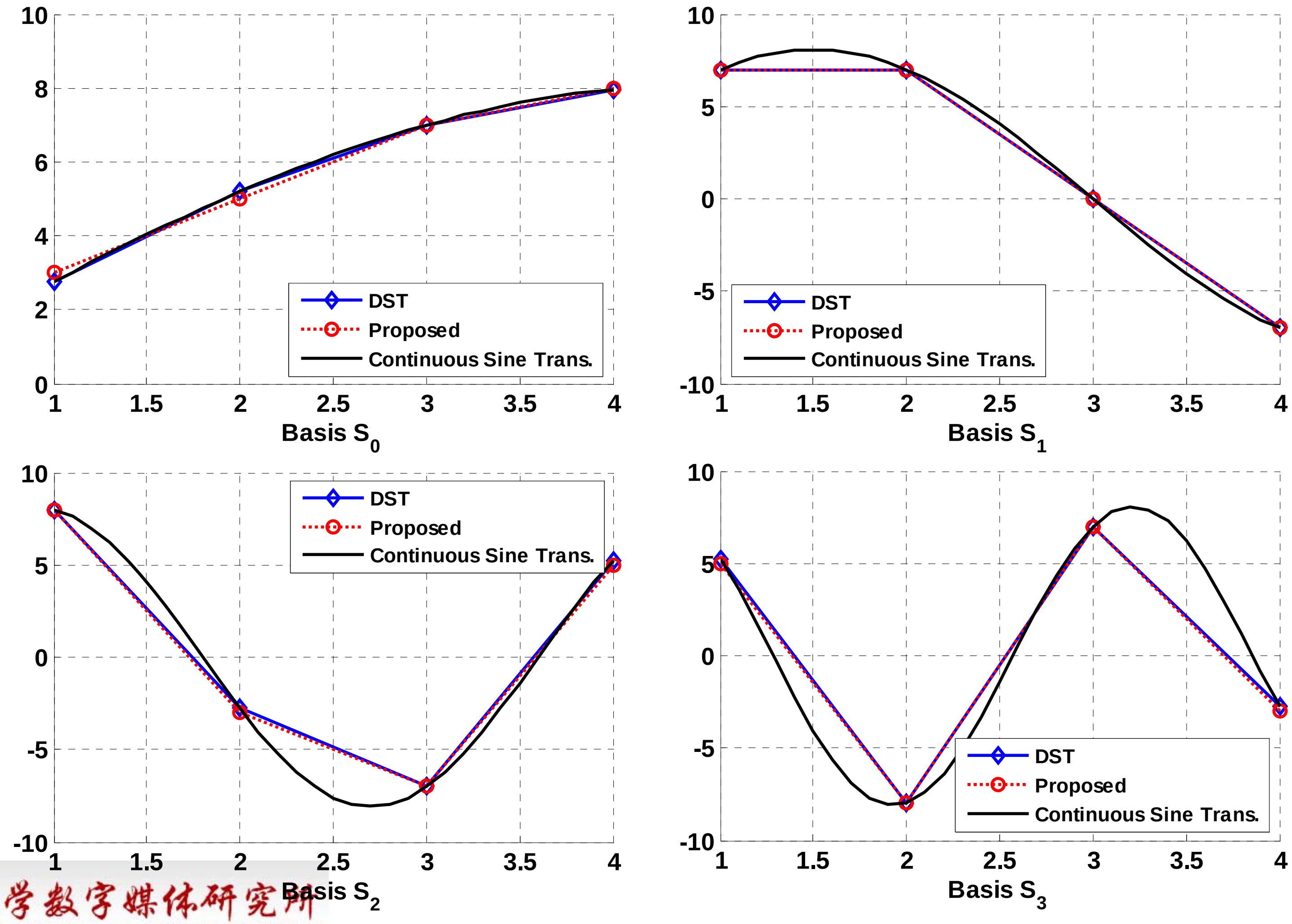
$$S_4 = \begin{bmatrix} 3 & 5 & 7 & 8 \\ 7 & 7 & 0 & -7 \\ 8 & -3 & -7 & 5 \\ -5 & 8 & -7 & 3 \end{bmatrix} \qquad S_4 \bullet S_4^T = \begin{bmatrix} 147 & 0 & 0 & 0 \\ 0 & 147 & 0 & 0 \\ 0 & 0 & 147 & 0 \\ 0 & 0 & 0 & 147 \end{bmatrix}$$

- The proposed 4x4 matrix is actually an approximation of 4x4 DST proposed by I^{2}R and Samsung

$$C_{i,j} = \frac{2}{\sqrt{2N+1}} \sin\left(\frac{(2i-1)j\pi}{2N+1}\right)$$

Algorithm

- Mode-dependent transform



Algorithm

- Mode-dependent transform
  - 4-point forward transform

$$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 3 & 5 & 7 & 8 \\ 7 & 7 & 0 & -7 \\ 8 & -3 & -7 & 5 \\ -5 & 8 & -7 & 3 \end{bmatrix} \bullet \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\begin{aligned} a_0 &= x_0 - x_1 \\ a_1 &= x_1 + x_3 \\ a_2 &= (x_2 << 3) - x_2 \\ a_3 &= x_0 + x_1 - x_3 \\ b_0 &= a_0 << 2 \\ b_1 &= a_1 << 2 \end{aligned}$$

$$\begin{aligned} c_0 &= b_0 + a_2 \\ f_0 &= -a_0 + c_0 + (a_1 << 3) \\ f_1 &= (a_3 << 3) - a_3 \\ f_2 &= (a_0 << 3) - a_2 + b_1 + a_1 \\ f_3 &= -a_0 - c_0 + b_1 - a_1 \end{aligned}$$

  - 4-point inverse transform

$$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 3 & 7 & 8 & -5 \\ 5 & 7 & -3 & 8 \\ 7 & 0 & -7 & -7 \\ 8 & -7 & 5 & 3 \end{bmatrix} \bullet \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\begin{aligned} a_0 &= x_0 + x_2 \\ a_1 &= x_2 - x_3 \\ a_2 &= (x_1 << 3) - x_1 \\ a_3 &= x_0 - x_2 - x_3 \\ b_0 &= a_0 << 2 \\ b_1 &= a_1 << 2 \end{aligned}$$

$$\begin{aligned} c_0 &= b_0 + a_2 \\ f_0 &= -a_0 + c_0 + b_1 + a_1 \\ f_1 &= a_0 + c_0 - (a_3 << 3) \\ f_2 &= (a_3 << 3) - a_3 \\ f_3 &= (a_0 << 3) - a_2 - b_1 + a_1 \end{aligned}$$

Algorithm

- Mode-dependent transform
  - S_8 is analytically obtained as

$$S_8 = \begin{bmatrix} 19 & 31 & 40 & 47 & 52 & 54 & 54 & 51 \\ 47 & 60 & 56 & 35 & 5 & -26 & -49 & -56 \\ 58 & 48 & 1 & -47 & -60 & -29 & 24 & 58 \\ 62 & 12 & -53 & -45 & 27 & 62 & 12 & -53 \\ 56 & -30 & -53 & 36 & 48 & -42 & -44 & 47 \\ 46 & -58 & 3 & 56 & -49 & -14 & 62 & -37 \\ 33 & -61 & 55 & -20 & -27 & 59 & -58 & 26 \\ -17 & 38 & -54 & 62 & -62 & 52 & -35 & 13 \end{bmatrix}$$

Algorithm

- Mode-dependent Coefficient Scanning

	Logical prediction modes (0~33)
Scanning Method #0	5~13, 21~29 (transposed version)
Scanning Method #1	1~4, 14~20, 30~33

Simulation results

- Intra High Efficiency

	with SMDS			no SMDS		
Class A	-2.5	-2.7	-2.5	-1.7	-1.6	-1.7
Class B	-1.2	-1.6	-1.5	-0.7	-0.7	-0.7
Class C	-1.7	-1.7	-1.6	-0.8	-0.5	-0.5
Class D	-1.8	-1.6	-1.7	-1.0	-0.7	-0.6
Class E	-2.7	-2.5	-2.3	-1.1	-1.1	-1.0
Average	**-1.8**	**-1.9**	**-1.8**	**-1.0**	**-0.8**	**-0.8**

Complexity

Complexity Metric	**4×4 Transform**
Operation counts for transform a *N*-point 1-D vector	DC: no change Others: 15 adds, 6 shifts
Memory requirements for transforms	no change
Memory requirements for coefficient scanning	2×4×16/8=16 byte (4 bits for one scan position, 2 for *x*, *2 for y)*
Minimum bit-precision (9-bit input)	DC: no change Others: 19bits (forward)
Memory for scaling matrices	DC: 6 matrices Others: 6 single values

Conclusion

- The coding efficiency is promoted using the proposed algorithm with applicable additional complexity
- Recommend further study on low-complexity mode-dependent transform and scanning methods

Thanks!