

HEVCにおける64x64CUの符号化効率改善

Coding efficiency improvement of 64x64 CU on HEVC

市ヶ谷 敦郎† 岩村 俊輔† 境田 慎一†
Atsuro Ichigaya Shunsuke Iwamura Shin-ichi Sakaida

1. まえがき

新放送サービスとして研究開発された8Kスーパーハイビジョン(以下SHV: Super Hi-Vision)がデジタルシネマから派生した4Kと共に2012年にITU-R Rec.BT2020 UHDTV(Ultra-High Definition Television system)として標準化された[1]。日本では世界に先駆けて2018年にBSによるUHDTV放送の開始を目指し、現在業界を挙げた準備が進められている。この放送では2013年に標準化されたHEVC符号化方式が利用される(2015年3月よりCSではChannel4Kで試験放送開始されている)。

HEVCはISO/IEC JTC1/SC29/WG11とITU-T SG16 Q.6によって共同で検討が進められ、2013年にMPEG-H part2/H.265 [2] [3]として標準化された最新の映像符号化方式であり、デジタル放送で用いられているMPEG-2の約4倍の圧縮性能を示す8Kに対応した最初の符号化方式となった。現在CSで行われている4K試験放送において1トランスポンダを使用しておよそ30-40Mbpsで良好な画質が得られている。また、HEVCを使用する場合、8K放送ではおよそ80-90Mbpsの帯域が必要であると報告されており、こちらもBSで1トランスポンダを使用して放送する見込みである[4]。一方、チャンネルの有効利用やより帯域の狭い地上放送における高画質な放送実現のためにはより高圧縮な符号化方式の開発が期待されている。本稿では64x64TUの導入による符号化効率改善の可能性と並列処理と親和性の高い実装による新しい変換処理について報告する。以下2章ではHEVC符号化方式の概要を変換処理を中心に説明し、3章で64x64DCT導入効果を検証し、4章で提案する変換処理について述べ、DCTと性能を比較し、5章でまとめを述べる。

2. HEVC符号化方式の概要

2.1 マルチブロック構造と変換処理

HEVC符号化方式はMPEG-2やMPEG-4 part10(以下AVC: Advanced Video Coding)/H.264等と同じブロックベース符号化で主に変換と予測、エントロピー符号化で構成されている。従来のMPEG-2やAVCでは入力信号の解像度に関わらずマクロブロックと呼ばれる符号化ブロックは16x16に固定されている。そのため高解像度映像信号にとってはブロックサイズが小さく、低解像度映像信号にとっては16x16が大きく、信号性質に適応した符号化処理の制御が行えなかった。HEVCではマクロブロックの概念を拡張したCU(Coding Unit)と呼ばれる符号化ブロックを最小符号化単位として採用している。CUはマクロブロックと同様、動き補償予測やイントラ予測を制御する単位であり、

マクロブロックとの違いは最大64x64から8x8まで選択することが可能な点である。本枠組みの導入により解像度に応じた符号化ブロックサイズが採用可能なだけでなく、エッジやテクスチャ部あるいは平坦部など、局所的な信号性質に応じたブロックサイズの選択ができ、より入力信号に適した符号化処理が可能となっている。

HEVCではさらにPU(Prediction Unit)と呼ばれる予測を行うブロック、TU(Transform Unit)と呼ばれる変換を行うブロックが採用されており、それぞれが複数のブロックサイズを選択可能である。これらのブロックサイズの最適化により予測、変換を適用する信号範囲が最適になり、符号化効率の最大化が実現されている。

PUは最大64x64のブロックが選択可能であり、PUとCUの両方が64x64となる場合もある。このような大サイズブロックは高解像度画像において特に選択される傾向がある。動き補償予測は処理量が多いがさまざまな高速化が提案されており、64x64PUにおいても実時間処理が可能であるとの見込みで採用されている。例えばHEVC Reference software (HEVC Test Model (HM) 16.2)では高速な動き探索手法としてダイヤモンドサーチを用いたTZ(Test Zone) Search[5]の改良版が採用されている。これらの手法ではサブサンプリングした映像を利用するなど処理時のデータの効率的な削減によって高速化を実現した。

HEVCの変換は従来の画像圧縮処理で高い効果を示してきたDCTをベースに開発された。変換処理は検討当初は4x4および8x8についてはAVCと同じ実装が検討され、係数ごとに重みのついた整数DCTが採用された。一方、16x16以上の変換では回路規模が大きくなることからChenのバタフライ演算[6]を用いた実装が検討された。しかし、標準化の過程で、回路の再利用性を考慮し、4x4および8x8の変換においてもChenのバタフライ演算アルゴリズムを採用し、統一的な設計とすることとした。また、イントラ符号化においてイントラ予測残差特有の信号性質に適した整数DSTがイントラ4x4TUでのみ採用された。一方でバタフライ演算による複数回の乗算による演算精度の低下を抑えるために一部の演算をマトリクス計算に置き換えるパーシャルバタフライ演算を採用した(図1)[6]。また64x64TUの導入効果を評価したところ1%程度のビット削減効果しか示さなかったこともあり[8] [9]、実時間処理の実現性の困難さを鑑みHEVCの規格として64x64TUの採用は見送られ、最大TUサイズは32x32となった。一方で映像の精細感の再現性のためには4x4TUが重要であるとの結果より、最小TUは4x4と規定された。図2に各ブロックの分割例を示す。分割されたCUを根としPU、TUが配置され、必要に応じてそれぞれが再分割(四等分)される。

†NHK放送技術研究所

〒157-8510 東京都世田谷区砧 1-10-11

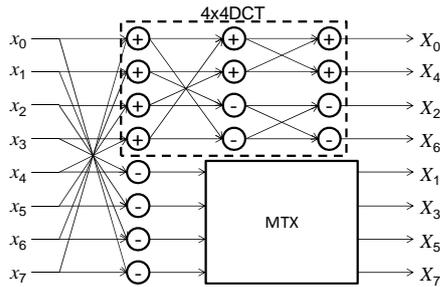


図1. パーシャルバタフライ演算(8点DCTの例)

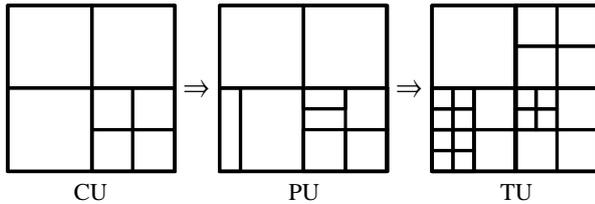


図2 CU, PU, TU分割例

2.2 変換係数の符号化

HEVC の変換係数の符号化では AVC での実績とその符号化効率の高さより算術符号化として CABAC が採用されている。複数のサイズの TU が入れ子状に分割されるため、TU の分割と有意係数を含む TU であるかどうかを示すフラグにより TU の分割情報を符号化する。それぞれの TU 内の変換係数はさらに 4×4 の係数グループ (Coefficients group : CG) に分けられ符号化される。TU 内の CG は図3に示す Diagonal scan によってスキヤニングされ、CG 内の変換係数も同様のスキヤニングによって有意係数の位置情報と値情報の2段階の情報によって符号化される。イントラ符号化においては図3に示す Diagonal scan に加えイントラ予測のモードに応じて水平優先および垂直優先の Horizontal および Vertical scan のいずれかが選択される。

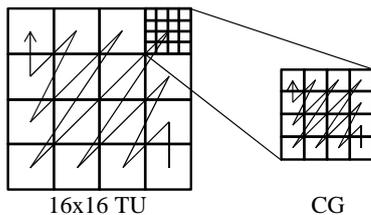


図3 係数スキヤニングの例(Diagonal scan)

3. 64x64TU の効果

HEVC に 64×64 TU を導入した場合の効果は HM で検証した。同様の結果が ITU-TSC16/Q.6 において報告されているが、この報告で用いられた符号化方式では TU を 64×64 に拡張し、低域 32×32 で帯域制限をかけ性能を評価している。また、動き予測関連、変換、ループフィルタなど多種の改善手法が同時に実装され、約 10% の改善が行えることを報告している[10]。しかし、これらの実験では HD 以上の解像度のシーケンスが少なく、 64×64 TU 導入の効果が不明である。そこで、現在の HM に 64×64 TU のみを追加した場合の性能改善効果を 4K シーケンスを用いて検証した。

3.1 64×64 変換

64×64 TU の変換処理は HEVC 検討段階で提案された 64 点整数 DCT 基底を用いて検証した[11]。本方式は 64 点 DCT 以外はすべて HEVC に採用された方式であり、64 点 DCT も 32 点以下の変換処理と同一コンセプト(パーシャルバタフライ演算)で設計された方式である。それゆえに HEVC の現在の変換処理が持つ揃えられた基底ノルムや N 点 DCT が $N/2$ 点 DCT を内包する構造など共通の特徴を備えている。エントロピー符号化については 64×64 係数に関する処理は HM に実装されていないため次のように行った。 64×64 TU を 4 つの係数グループに分割し、それぞれを 32×32 TU とみなすサブ TU とし、4 つの 32×32 サブ TU をそれぞれ 32×32 DCT 係数とみなし、スキヤニング、エントロピー符号化処理を実行し符号化を行った。また、 $4:2:0$ 信号の場合、 64×64 の輝度信号に対応する色差信号は 32×32 であり、輝度信号の 32×32 点 DCT と同じ処理を適用することも可能であるが、輝度と色差の処理の対称性を考え、4 つの 16×16 サブ TU とし、こちらも 16×16 TU と同様に処理することとした。4 つのサブ TU はそれぞれ有意係数を持つかどうかを示す coded_block_flag (cbf) を導入した。

処理手順

- ・輝度信号を 64×64 TU に分割し、 64×64 DCT を実行
- ・ 64×64 の係数を 32×32 サブ TU に分けそれぞれのサブ TU に有意係数が含まれるか否かで cbf を設定
- ・ 32×32 サブ TU 内の変換係数は 32×32 TU と同等の手順で符号化 (コンテキストは各ブロックで独立)
- ・輝度信号 64×64 TU に対する色差信号 32×32 TU は 32×32 DCT による変換処理後、4 つの 16×16 サブ TU に分割し、係数を符号化

3.2 符号化実験

本変換処理を HEVC Reference software (HEVC Test Model (HM) 16.2) に実装して評価を行った。テスト素材として撮影した 8K 映像を 4K にダウンコンバートした 10 シーケンスの約 1 秒分(64 フレーム)を使用した[4]。サムネイルを図4に示す。また、符号化条件は JCT-VC で用いられた Common Test Condition (CTC)[12] を用いており、固定量子化によって符号化を行った。使用した符号化条件の概要を表1に示す。



001 Beach



002 Cracker



003 TownDolly



004 Snow



図 4 テスト画像

表 1 符号化実験の条件

映像信号形式	Y'CbCr 4:2:0, 10 bit
解像度	3840x2160
表色系	Rec. ITU-R BT.709
エンコーダ	HEVC Test Model HM16.2 +64x64TU 拡張
HEVC プロファイル	Main 10
量子化パラメータ(QP 値)	22, 27, 32, 37
符号化構造	All Intra (AI), Random Access (RA), Lowdelay B (LB), Lowdelay P (LP)

3.3 実験結果

符号化結果を BD-BitRate によって評価した。評価に用いた指標は輝度信号の BD-BitRate であり、比較対象と同等の品質を実現するために必要とするビット量が比較対象に対してどれくらいかを示す指標である。そのため負値が性能の改善を示している。比較対象として HM16.2 の符号化結果を用いた。実験結果を表 2 に示す。

表 2 評価結果 (BD-BitRate)

	Main10			
	AI	RA	LB	LP
001_Beach	0.0%	-0.1%	0.1%	0.0%
002_Cracker	0.0%	-0.1%	0.0%	0.0%
003_TownDolly	-0.1%	-0.2%	0.0%	-0.1%
004_Snow	0.0%	0.0%	-0.1%	0.0%
005_Flame	-0.1%	0.0%	0.1%	-0.1%
006_Diving	0.0%	0.1%	0.1%	0.0%
007_Nebuta	0.0%	-0.6%	0.0%	0.1%
008_Farm	0.0%	0.0%	0.0%	0.0%
009_Riverboat	-0.2%	0.0%	0.1%	0.0%
010_Twilight	-0.1%	-3.7%	-5.9%	-6.9%
Average	0.0%	-0.3%	-0.5%	-0.6%

表 2 に示す通り、全体としてイントラ符号化において 64x64TU の効果がほとんどないことが分かる。一方、インター符号化フレームの含まれる RA, LB, LP 設定において符号化効率の改善効果が高い。シーケンスごとに分析すると正值を示すものは多くても 0.1% であり、平均では符号化効率の改善を示す負値となっており、最大 LP 条件で平均 0.6% の改善を示している。特に 010_Twilight においては最大 6.9% のビット量削減効果を示した。010_Twilight は夕暮れの街並みを間欠撮影によって暗くなる様子を撮影したシーケンスであり、サムネイルに示す通り画面の大部分を空が占めており、定点カメラによる夜更けの様子を撮影している。空が大部分を占める絵柄としては 001_Beach および 009_Riverboat もあるがこれらとの違いは 010_Twilight は大面積で輝度の変化がある点である。001_Beach や 009_Riverboat では空の領域で 64x64CU が選択されやすいが信号の変化に乏しいためほとんどが予測によって再構成され、予測差分信号が発生しない。そのため有意係数が 1 つも発生せず 64x64TU が選択されていなかった (Skip 処理)。一方で 010_Twilight も空の領域で 64x64CU が選択されていたが、001_Beach や 009_Riverboat と異なりフレームごとの輝度のレベルに変動が生じ、予測差分成分が発生しており 64x64TU に有意係数が発生し、64x64TU の導入効果が顕著であった。

4. 64x64TU の改善

これらの結果から、64x64TU は特定の映像において顕著な効果を示すことが確認できた。8K 信号においては 64x64TU は HD における 16x16TU に相当するブロックサイズであり、今後の 8K コンテンツの充実に伴いその重要度は増すものと考えられる。しかし、HEVC の検討段階で実装上の困難さが指摘されており実装方法の改善が必要である。そこで 64x64 変換ブロックを並列化しやすい実装で変換処理する方法を新たに提案する。提案する変換処理は MPEG や ITU-T で検討が開始された次世代の新符号化方式の検討で議論されている要求条件案も参考にした。提案する変換処理の特徴を以下に示す。

- 64x64 のブロックを並列処理しやすい構造をもつこと
- 処理遅延が少ないこと
- スケーラブル機能を特別な機能の付加なく行えること

前述の通り HEVC の変換処理には Chen のバタフライ演算をベースにした処理が採用された。Chen のバタフライ演算は変換核の対称性を利用した高速処理であり、図 1 に示すように演算過程の第一段目において x_k と $x_{(N-1)-k}$ の加減算が生じる。そのため垂直方向の変換処理のために処理の開始に $N/2$ ラインのデータ読み込み遅延が必要となる。つまり、 N 点変換のためには水平画素数 W に比例して処理の開始までに最少でも $W \times N/2$ 画素を読み込む必要があり、水平解像度 W とブロックサイズ N に比例して処理遅延も増大する。そこで本検討では並列処理を実現するために読み込んだ信号を逐次処理可能な実装を提案する。

4.1 提案方式

DCT の再帰的な処理を実現はこれまでに様々な手法が報告されている。HEVC に採用された Chen のバタフライ演算はもっとも基本的なバタフライ演算であるが、前述した

ように特に垂直方向の変換処理の実行に処理遅延が発生する。そこで、逐次処理が行えるように隣接する画素ごとに処理を行う変換処理を設計した。また、64x64 変換を軽量に行う工夫として画素を 2x2 ウォルシュ型のアダマール変換によって平均画像と、エッジ (微分) 画像に変換し、それぞれ 32x32TU として処理することとした。アダマール変換は式 1 であらわされる変換であり、乗算を用いないため軽量であり、変換画像は隣接画素を加算平均した平均画像と隣接画素の差分による微分画像を出力するという特徴を持つ。

(式 1)

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

NxN 点入力画像のアダマール変換は、低域成分は画素平均による低解像度縮小画像であり、従来の映像信号と類似した性質をもつ。そこで低域成分については N/2 点 DCT によって符号化を行うこととした。また、高域成分は微分画像であり、高周波数成分を多く含む画像となっている。高周波信号に対する変換では DST のエネルギー集中度が高いことが知られており、N/2 点 DST により変換処理を行うこととした。

N 点 DST は式 2 に示す変換処理であり、DCT における変換核を余弦関数から正弦関数にしたものである。

(式 2)

$$X_s(k) = \sqrt{\frac{2}{N}} c_k \sum_{n=0}^{N-1} x(n) \sin\left(\frac{(2n+1)k\pi}{2N}\right) \quad u = 1, \dots, N$$

ただし、 $c_k = \begin{cases} \frac{1}{N} & k \neq N \\ \sqrt{\frac{1}{N}} & k = N \end{cases}$

DST は正弦関数の変形により式 3 に示す余弦関数の変換核に変換可能であり、下式に示すように DCT によって表現される。 [13]

(式 3)

$$X_s(k) = \sqrt{\frac{2}{N}} c_k \sum_{n=0}^{N-1} (-1)^n x(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad u = 0, \dots, N-1$$

ただし、 $c_k = \begin{cases} \frac{1}{N} & k \neq 0 \\ \sqrt{\frac{1}{N}} & k = 0 \end{cases}$

つまり、提案する変換処理は N/2 点 DCT を利用することにより、回路の再利用性が高い処理である。提案する変換処理の信号線図を図 5 に示す。DST を構成する DCT の入力信号の -1 倍は減算の順序を入れ替えるのみであるため本質的な処理量の増加は無いとみなせ、提案方式の追加される処理量は 32x32 変換処理に加え、前処理の NxN 回の加算処理のみであり、64x64DCT に比べ軽量である。

本提案手法は x_k, x_{k+1} 番目の入力信号の入力があれば逐次加算、減算処理を開始することができ、特に垂直方向の変換処理において信号読み出しによる処理遅延は 64x64DCT の $W \times N/2$ に対し、ブロックサイズ N に関わらず W ですみ、処理遅延の大幅な削減を実現する。また、

アダマール変換後の信号処理は完全に独立して行えることから並列処理の実現も容易な構成となっている。逆変換において低域側の逆 DCT 出力は水平・垂直 1/2 解像度の低解像度画像を直接出力でき、簡易なスケラブル機能も実現可能である。

本提案手法では 64x64 点の映像信号をアダマール変換によって 4 つの 32x32 の低解像度画像と微分画像に分解し、それぞれに変換処理を行う。これらの要素画像の 32x32 係数の単位をサブ TU とし、64x64DCT と同様、それぞれのサブ TU に有意係数があるかどうかを示す cbf を導入した。サブ TU 内の有意係数は従来の 32x32TU と同様の処理とした CABAC のコンテキストはブロックごとに独立とした。

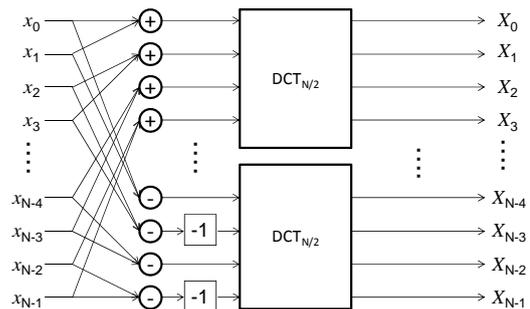


図 5 提案する N 点変換処理

4.2 提案法の性能評価および考察

符号化実験を 3.2 と同一の条件で行った。結果を表 3 に示す。表 2 同様符号化性能を BD-Bitrate で示す。

表 3 評価結果 (BD-Bitrate)

	Main10			
	AI	RA	LB	LP
001_Beach	0.0%	0.0%	0.0%	0.0%
002_Cracker	0.0%	0.0%	0.0%	0.0%
003_TownDolly	0.0%	0.0%	0.0%	0.1%
004_Snow	0.0%	0.0%	0.0%	-0.1%
005_Flame	0.4%	0.3%	0.3%	0.3%
006_Diving	0.0%	0.0%	0.0%	0.0%
007_Nebuta	0.0%	0.0%	0.0%	0.0%
008_Farm	0.0%	0.0%	0.0%	0.0%
009_Riverboat	-0.1%	0.1%	0.2%	0.2%
010_Twilight	0.0%	-3.6%	-5.9%	-6.8%
Average	0.0%	-0.3%	-0.5%	-0.6%

表 2 と比較すると、005_Flame において 0.3-0.4% のビット量の増加を生じており、符号化効率の若干の悪化が生じている。一方で 64x64DCT による TU 分割で効果を示した 010_Twilight において、おおむね提案方式も同程度の改善を示している。両者を 64x64TU の選択率によって比較する。005 における 64x64TU の選択領域の面積は AI 条件でもっとも選択率が高い QP=37 においてもわずか 1.1% と極めて小さく、ほとんどの条件において 1% 以下であった。同条件の 64x64DCT の場合は 5.5% であり、およそ選択率が 1/5 以下に低下していた。これは、複数の 32x32 変換によりエネルギーが分散し、エントロピー符号化効率が低下したため選択率が低下したものと考えられる。そのためほとんど選択されず、導入された制御情報分がそのままビット量の悪化につながったと考えられる。一方、010_Twilight は最も性能向上を示した LP 条件 QP=22 において 9.4% と選択率

が高く、導入効果があると考えられる。これらの結果より、提案する変換処理が 005_Flame の絵柄に適していなかったことが推測される。今回の実装では各サブブロックで共通の量子化、スキヤニングを行っており、サブブロック間の依存性や周波数特性の違いを利用していない。これらを考慮し、最適化を図ることにより性能の改善が期待される。

5. まとめ

本稿では既存の HEVC 符号化方式に 64x64TU を導入し、4K シーケンスによる符号化効率改善の可能性を実験により明らかにし、最大 6.9%の符号化効率の改善が可能であることを示した。また、従来の変換処理に対し、より並列処理と親和性の高い変換処理を提案し、その性能を明らかにした。提案手法はイントラブロックにおいて最大 0.4%の符号化効率の低下が生じたものの、最大 6.8%と 64x64DCT と同等の改善を示すこと確認した。性能低下は、今後処理の最適化によって改善可能であると考えている。提案手法はアダマール変換による低解像度画像と微分画像に対する 32x32 変換処理で実現できることからそれぞれの画像に対する量子化制御をおこなうことによって従来の符号化では劣化が発生しやすいエッジ情報の制御が行え、復号画像の高画質化が期待される。また、回路の再利用性が高く、高い並列処理が実現でき、実用性が高い変換処理であると考えられる。本検討では予測方式の変更を行わなかったが、符号化方式の入力段においてアダマール変換による要素画像に分解することにより、処理の重い動き探索などに低解像度画像を利用することで 4K や 8K などの高解像度映像信号の処理量の低減にも応用できると考えられる。これらについても今後検討を進めていく。

参考文献

- [1] ITU-R Recommendation BT.2020, "Parameter values for ultra-high definition television systems for production and international programme exchange," 2012.
- [2] ISO/IEC 23008-2, "Information technology -- High efficiency coding and media delivery in heterogeneous environments -- Part 2: High efficiency video coding," 2013
- [3] ITU-T Recommendation H.265, "SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video High efficiency video coding," 2013.
- [4] 市ヶ谷, 西田, "HEVCにおける 8K, 4K, 2K 映像符号化の所要ビットレート評価" 英情学技報, vol. 38, no.8 BCT2014-44, pp.47-52, (2014/12)
- [5] N.Purnachand, L.Nero Alves, A.Navarro, "Improvements to TZ searchmotion estimation algorithm for multiview video coding", IEEEIWSSIP 2012, Vienna Apr 2012.
- [6] W.H. Chen, C.H. Smith, S. Fralick "A fast computational algorithm for the discrete cosine transform" IEEE Trans. Comm., 25 (1977), pp. 1004–1009
- [7] P. Topiwala, M. Budagavi, R. Joshi and E. Alshina, "CE10: Summary report on Core transform design", JCTVC-G040/m21829
- [8] M. Zhou, "Proposed low complexity video encoder settings for HEVC", JCTVC-B054/m24141
- [9] M. Zhou, "TE12: Evaluation of transform unit (TU) size", JCTVC-C056/m18079
- [10] "Coding tools investigation for next generation video coding", ITU-T COM16-C806-E
- [11] A. Fuldseth, G. Bjøntegaard, "Unified transform design for HEVC with 16 bit intermediate data representation," JCTVC-D224/ m18984
- [12] F. Bossen, "Common HM test conditions and software reference configurations," JCTVC-L1100/m28412
- [13] Z. Wang, "A fast algorithm for the discrete sine transform implemented by the fast cosine transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 814.815, Oct. 1982.