ウォルシュ変換における音像・画像圧縮技術の一検討

1Q02

今井　幸雄
東海大学短期大学部情報・ネットワーク学科　〒108-8619 東京都港区高輪 2-3-23
Tel03-3441-1171 Fax03-3447-6005 E-mail:imai@ttc.u-tokai.ac.jp

まえがき　マルチメディア技道を導入した創造化マルチメディアの時代ではシステムＶＬＳＩ（種々の演算チップ・ＣＰＵチップ）の設計・試作が活発に研究されている。それを成し遂げる一例として、ＨＤＬ（Hardware Description Language）を用いてＦＰＧＡで実現する。それらを使って画像処理プログラム等の研究が提案され、実用化されている。
マルチメディアを扱う技術として変換技術がある。すでにJPEG、MPEGではDCT技術が標準化技術として導入され、利用されている。ウォルシュ変換技術はコンピュータの処理技術に最適であり高速、高能率技術であることからその変換技術を圧縮技術に適用考察している。高速ウォルシュ変換技術と同種の高速ハールウェブレット変換技術についても検討している。カオスシステム、超音波音像・画像システム、電流帰還形演算増幅器システム等のマルチメディア処理システムのLSIハードウェア設計も検討している。
高速ウォルシュ変換　ウォルシュ変換を高速化する算法を考える。FFTアーキテクチャーと同様高速ウォルシュ変換バタフライを導入する。乗算演算が無く、加算演算回数がNlogNと少ない回数の演算でウォルシュ変換が可能である。
３Ｄ画像（映像）圧縮シミュレーションの設計　１Ｄ画像（音像）のウォルシュ変換、２Ｄ画像のウォルシュ変換の拡張として、３Ｄ画像（映像）のウォルシュ変換についての理論的考察を行う。
２Ｄ画像のウォルシュ変換は画像データに左右からウォルシュ変換核とその転置変換核を乗じたものである。それにｚ軸を導入すると、３Ｄ画像のウォルシュ変換が得られる。はじめにｘｙ平面に関する変換をｚ軸について実行すると

$$\sum_{z=0}^{N-1}(\mathbf{W})\mathbf{a}_{x(z)y(z)z}(\mathbf{W})^t=\mathbf{A}_{xyz}$$

次に上の出力変換データを入力データとして、ｚｘ平面に関する変換をｙ軸について実行すると

$$\sum_{y=0}^{N-1}(\mathbf{W})\mathbf{a}_{x(y)yz(y)}(\mathbf{W})^t=\mathbf{A}_{xyz}$$

“Consideration of image compression technology in Walsh transform”, Yukio Imai, Tokai Univ., Jun., Coll., 2-3-23, Takanawa, Minato-ku, Tokyo, 108-8619 Japan

最後に上の出力変換データを入力データとして、ｙｚ平面に関する変換をｘ軸について実行すると

$$\sum_{x=0}^{N-1}(W)a_{xy(x)z(x)}(W)^t=A_{xyz}$$

となり、３Ｄ画像のウォルシュ変換が得られる。
ウォルシュ変換プログラム　シミュレーション実験例として、画像処理するプログラムを次に示す。
あとがき　高速ウォルシュ変換の算法を述べた。その変換は乗算が無く、加算回数が少なくて済む。変換後は低周波に対応する部分だけを残し、高周波に対応する部分はゼロとして圧縮を考える。高速ウォルシュ逆変換も高速ウォルシュ変換と同じ操作である。ウォルシュ変換プログラムで確認し、良好な結果を得ることができた。ＤＣＴⅡ＆ＩＤＣＴⅡを用いた圧縮についても考察した。そこで用いたＤＣＴⅡ＆ＩＤＣＴⅡの定義式は次式である。ＤＣＴⅡ：

$$\mathbf{X}_m^N=\sqrt{\frac{2}{N}}\mathbf{k}_m^N\sum_{n=0}^{N-1}\left(\mathbf{x}_n^N\cos\left(\frac{(2n+1)m\pi}{2N}\right)\right)$$

ＩＤＣＴⅡ：

$$\mathbf{x}_n^N=\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\left(\mathbf{k}_m^N\mathbf{X}_m^N\cos\left(\frac{(2n+1)m\pi}{2N}\right)\right)$$

$$m,n=0\sim N-1$$

ただし、$k_j^N=1\ldots(j=1,2,\ldots,N-1)$

$$k_j^N=\frac{1}{\sqrt{2}}\ldots(j=0)$$

Ｎが８、１６の場合の高速コサイン変換について検討する。ＤＣＴⅡを行列の形式で表現すると

$$\begin{bmatrix}X_0\\X_1\\X_2\\X_3\\X_4\\X_5\\X_6\\X_7\end{bmatrix}=\sqrt{\frac{2}{8}}\begin{bmatrix}\frac{1}{\sqrt{2}} & & \frac{1}{\sqrt{2}}\\ \cos\frac{\pi}{16} & & \cos\frac{15\pi}{16}\\ & & \\ & & \\ & & \\ & & \\ & & \\ \cos\frac{7\pi}{16} & & \cos\frac{105\pi}{16}\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{bmatrix}$$

cos の部分を省略した変換核行列は

$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$\frac{\pi}{16}$	$\frac{3\pi}{16}$	$\frac{5\pi}{16}$	$\frac{7\pi}{16}$	$\frac{9\pi}{16}$	$\frac{11\pi}{16}$	$\frac{13\pi}{16}$	$\frac{15\pi}{16}$
$\frac{2\pi}{16}$	$\frac{6\pi}{16}$	$\frac{10\pi}{16}$	$\frac{14\pi}{16}$	$\frac{18\pi}{16}$	$\frac{22\pi}{16}$	$\frac{26\pi}{16}$	$\frac{30\pi}{16}$
$\frac{3\pi}{16}$	$\frac{9\pi}{16}$	$\frac{15\pi}{16}$	$\frac{21\pi}{16}$	$\frac{27\pi}{16}$	$\frac{33\pi}{16}$	$\frac{39\pi}{16}$	$\frac{45\pi}{16}$
$\frac{4\pi}{16}$	$\frac{12\pi}{16}$	$\frac{20\pi}{16}$	$\frac{28\pi}{16}$	$\frac{36\pi}{16}$	$\frac{44\pi}{16}$	$\frac{52\pi}{16}$	$\frac{60\pi}{16}$
$\frac{5\pi}{16}$	$\frac{15\pi}{16}$	$\frac{25\pi}{16}$	$\frac{35\pi}{16}$	$\frac{45\pi}{16}$	$\frac{55\pi}{16}$	$\frac{65\pi}{16}$	$\frac{75\pi}{16}$
$\frac{6\pi}{16}$	$\frac{18\pi}{16}$	$\frac{30\pi}{16}$	$\frac{42\pi}{16}$	$\frac{54\pi}{16}$	$\frac{66\pi}{16}$	$\frac{78\pi}{16}$	$\frac{90\pi}{16}$
$\frac{7\pi}{16}$	$\frac{21\pi}{16}$	$\frac{35\pi}{16}$	$\frac{49\pi}{16}$	$\frac{63\pi}{16}$	$\frac{77\pi}{16}$	$\frac{91\pi}{16}$	$\frac{105\pi}{16}$

参考文献 [1] 今井幸雄、宅間俊則：“情報演算処理の高速化に関する２，３の考察”、情処学 55 回 H9 後全大、4F-01

[2] 貴家他：“DCT(離散コサイン変換)入門“、1997 年９月 20 日、ＣＱ出版(株)

圧縮処理のための行列積のプログラム例

```
#include <stdio.h>
#define NUM_N 3
#define NUM_L 4
#define NUM_M 5
void main(void)
{
    float a[NUM_N][NUM_L];
    float b[NUM_L][NUM_M];
    float c[NUM_N][NUM_M];
    float d[NUM_M][NUM_N];
    float e[NUM_N][NUM_N];
    int i,j,k;
    for(i=0;i<NUM_N;i++)          /*行列Ａの入力*/
    {
        for(j=0;j<NUM_L;j++)
        {
            printf("A[%d,%d]=",i+1,j+1);
            scanf("%f",&a[i][j]);
        }
    }
    for(i=0;i<NUM_L;i++)          /*行列Ｂの入力*/
    {
        for(j=0;j<NUM_M;j++)
        {
            printf("B[%d,%d]=",i+1,j+1);
            scanf("%f",&b[i][j]);
        }
    }
    for(i=0;i<NUM_M;i++)      /*行列Ｄの入力*/
    {
        for(j=0;j<NUM_N;j++)
        {
            printf("D[%d,%d]=",i+1,j+1);
            scanf("%f",&d[i][j]);
        }
    }
    for(i=0;i<NUM_N;i++)     /*Ｃ＝Ａ＊Ｂの計算*/
    {
        for(j=0;j<NUM_M;j++)
        {
            c[i][j]=0;
            for(k=0;k<NUM_L;k++)
            {
                c[i][j]+=a[i][k]*b[k][j];
            }
        }
    }
    printf("¥n");
    printf("C:¥n");               /*行列Ｃの表示*/
    for(i=0;i<NUM_N;i++)
    {
        for(j=0;j<NUM_M;j++)
        {
            printf("%10.5f",c[i][j]);
        }
        printf("¥n");
    }
    for(i=0;i<NUM_N;i++)     /*Ｅ＝Ｃ＊Ｄの計算*/
    {
        for(j=0;j<NUM_N;j++)
        {
            e[i][j]=0;
            for(k=0;k<NUM_M;k++)
            {
                e[i][j]+=c[i][k]*d[k][j];
            }
        }
    }
    printf("¥n");
    printf("E:¥n");               /*行列Ｅの表示*/
    for(i=0;i<NUM_N;i++)
    {
        for(j=0;j<NUM_N;j++)
        {
            printf("%10.5f",e[i][j]);
        }
        printf("¥n");
    }
}
```