2005－AVM－50（4）
2005／10／7

H.264/AVC の直交変換におけるパディング法

黒木　祥光 †　　廣重　徹 †　　上繁　義史 ††　　鎌田清一郎 †††

† 久留米工業高等専門学校
†† 九州システム情報技術研究所
††† 早稲田大学

あらまし　本論文は，任意形状画像の符号化においてしばしば用いられるパディングについて考察するものである．パディングは，一般に，任意形状領域内の画素を用いて領域外部を埋めて矩形ブロックを作成することを意味し，動画像符号化 MPEG4 においても，low-pass extrapolation (LPE) と称する手法が採用されている．1 次元 DCT に対するパディング法として，既に，Shen と Liou は，領域内画素数と符号化すべき DCT 係数の個数が等しいことを保証する手法を報告している．彼等は，同時に，画像信号への適用法も 2 種類提案しているが，彼等の手法は 1 次元 DCT を基本としているため，領域内画素と符号化すべき DCT 係数の個数が一致しない．本論文では，Shen らの手法を H.264/AVC で用いられる整数精度 DCT を用いた 2 次元 DCT に拡張する．提案する手法では，直交変換後の符号化処理を考慮し，領域内画素数と同数の DCT 係数がジグザグ順序の初めに出現するため，Shen らの手法に比べ，更なる符号量の削減が期待できる．

A padding technique for orthogonal transform on H.264/AVC

Yoshimitsu KUROKI†, Akira HIROSHIGE†, Yoshifumi UESHIGE††, and Sei-ichiro KAMATA†††

†Kurume National College of Technology
††Institute of Systems & Information Technologies/KYUSHU
†††Waseda University

Abstract　This paper presents a new padding technique for arbitrarily shaped coding. Padding background pixels using pixels in shapes, square blocks are generated; therefore, traditional orthogonal transforms work availably. A padding technique called low-pass extrapolation (LPE) is indeed employed in MPEG-4. Shen and Liou demonstrate a sophisticated padding for 1D-DCT, which guarantees that as many high frequency DCT coefficients as background pixels become zero. In order to apply their method to 2D-DCT, they also show two solutions. However, the solutions do not accomplish the equivalence between the numbers of the high frequency coefficients and the background pixels because the proposals are merely based on the 1D-DCT. In this paper, we enhance their padding methods to the orthogonal transform used in H.264/AVC, namely integer DCT. In the proposed method, the number of the DCT coefficients to be coded is equivalent to the number of the pixels lie in the shape and is appear in the zigzag order.

1　まえがき

任意形状画像の符号化は，画像の加工・編集を容易に実現するために必須の技術であり，今後，その重要度は益々増加するものと考えられる．実際，動画像符号化 MPEG4[1] では，任意形状画像の符号化として，shape-adaptive DCT (SA-DCT) あるいは low-pass extrapolation (LPE)，何れかの手法を用いることになっている．後者は，文字通り領域外を計算負荷の少ない一種の外挿フィルタで埋めて矩形ブロックを作成した後，標準の 2 次元 DCT を施す手法である．なお，この埋め合わせの手法は一般にパディングと呼ばれている．LPE の場合，作成された矩形ブロックは滑らかな画像になるため，DCT 係数が低周波に集中すると期待されるが，高周波成分が必ず 0 になるとは保証されない．

一方，Shen と Liou[2] は 1 次元 DCT に対し，領域外の画素数と同数の高周波成分が必ず 0 となるパディング法を提案している．彼らの手法は，あくまで 1 次元 DCT を基本としており，$N \times N$ の画像信号へ適用する際，以下の 2 種類の手法を提案している．1) 先ず N 個の独立した N 次元列ベクトルに対し，パディング及び 1 次元 DCT を施した後，N 次元の DCT 係数から成る行ベクトルに対し，同様の処理を行う．2) 先に「行－列」，あるいは「列－行」の順にパディングした後，2 次元 DCT を行う．Shen らは双方に対するシミュレーション結果も踏まえて考察した結果，手法 2) の方が有効であるとの結論に至っている．しかし，2 次元 DCT に先立ってパディングを行うため，DCT 係数の個数が領域内画素数と同じになる事は保証されない．また，符号化すべき

DCT 係数もジグザグ走査の順に配置されるとは限らないため，符号化の際，新たな走査順序を用意する必要がある．

著者らは先の研究で，Shen らの手法を 2 次元 DCT へ拡張する手法について考察し，領域内画素数と符号化すべき DCT 係数の個数が等しく，かつ，ジグザグ走査の順に出現するパディング法を示した [3]．しかし，パディングすべき画素の位置を全探索で求める場合，その組合せは最大で $\begin{pmatrix} 64 \\ 32 \end{pmatrix} \approx 1.83 \times 10^{18}$ 存在するため，未だ導出には至っていない．

ところで，最新の動画像符号化の国際標準 H.264/AVC の直交変換は処理の高速化や他の処理との整合性から，従来の国際標準より小さな 4×4 画素を単位としている [4]．従って，上述した画素位置の探索数は最大でも $\begin{pmatrix} 16 \\ 8 \end{pmatrix} = 12,870$ であるため，計算機を用いて導出することが可能となる．本論文では H.264/AVC の直交変換におけるパディングについて示す．

2　パディング行列の導出法 [2]

N 次元のベクトル $\boldsymbol{x} = {}^t(x_1\, x_2 \ldots x_N)$ のうち，$K\,(1 \leq K \leq N)$ 個の領域内ベクトル $\boldsymbol{x}_1$ を用いて $N - K = L$ 個の領域外ベクトル $\boldsymbol{x}_2$ をパディングすると仮定する．単純に $\boldsymbol{x}$ の上位に位置する K 個の要素から成るベクトルを $\boldsymbol{x}_1$，下位の L 個の要素から成るベクトルを $\boldsymbol{x}_2$ と仮定すると，$\boldsymbol{x} = {}^t(\boldsymbol{x}_1\, \boldsymbol{x}_2)$ より，$\boldsymbol{x}_1 = {}^t(x_1\, x_2 \ldots x_K)$，$\boldsymbol{x}_2 = {}^t(x_{K+1}\, x_{K+2} \ldots x_N)$ となる．変換行列を $\boldsymbol{A} = (a_{ij})\,(i,j = 1,\ldots,N)$ とし，変換係数 $\boldsymbol{y}$ を $\boldsymbol{x}$ と同様に K 次元のベクトル $\boldsymbol{y}_1$ と L 次元のベクトル $\boldsymbol{y}_2$ に分けると，$\boldsymbol{A}$ による $\boldsymbol{x}$ の線形変換は次式で表すことが出来る．

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix} \tag{1}$$

ここで，$\boldsymbol{A}$ を次に示す部分行列 $\boldsymbol{A}_{1\,1}$，$\boldsymbol{A}_{1\,2}$，$\boldsymbol{A}_{2\,1}$，$\boldsymbol{A}_{2\,2}$ に分割する．

$$\begin{aligned} \boldsymbol{A} &= \left(\begin{array}{c|c} \boldsymbol{A}_{1\,1} & \boldsymbol{A}_{1\,2} \\ \hline \boldsymbol{A}_{2\,1} & \boldsymbol{A}_{2\,2} \end{array}\right) \\ &= \left(\begin{array}{ccc|ccc} a_{1\,1} & \cdots & a_{1\,K} & a_{1\,K+1} & \cdots & a_{1\,N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{K\,1} & \cdots & a_{K\,K} & a_{K\,K+1} & \cdots & a_{K\,N} \\ \hline a_{K+1\,1} & \cdots & a_{K+1\,K} & a_{K+1\,K+1} & \cdots & a_{K+1\,N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{N\,1} & \cdots & a_{N\,K} & a_{N\,K+1} & \cdots & a_{N\,N} \end{array}\right) \end{aligned} \tag{2}$$

式 (1)，(2) を用いて $\boldsymbol{x}_2$ について解くと，$\boldsymbol{A}_{2\,2}$ が正則の場合，次式を得る．

$$\boldsymbol{x}_2 = \boldsymbol{A}_{2\,2}^{-1}(\boldsymbol{y}_2 - \boldsymbol{A}_{2\,1} \cdot \boldsymbol{x}_1) \tag{3}$$

$\boldsymbol{y}_2$ は任意のベクトルに設定可能であるが，ここでは高周波成分を 0 とすることが目的であるため，$\boldsymbol{y}_2 = \boldsymbol{0}$ とすると，最終的に，$\boldsymbol{x}_2$ は次式で表される．

$$\boldsymbol{x}_2 = -\boldsymbol{A}_{2\,2}^{-1}\boldsymbol{A}_{2\,1} \cdot \boldsymbol{x}_1 \tag{4}$$

以下，L 行 K 列の行列 $\boldsymbol{\Pi} = -\boldsymbol{A}_{2\,2}^{-1}\boldsymbol{A}_{2\,1}$ をパディング行列と呼ぶ．

更に，Shen らは領域内成分と領域外成分の選択法も示している．今，$\boldsymbol{x}$ の各成分の添字から成る集合を $R = \{1, 2, \ldots, N\}$ とする．また，領域内ベクトル $\boldsymbol{x}_1$ 各成分の添字の集合を $P = \{p_1, p_2, \ldots, p_K\}$ とすると，P は R の部分集合であり，$\boldsymbol{x}_2$ の各成分の添字の集合は P の R に関する補集合 $P^c = \{p'_1, p'_2, \ldots, p'_L\}$ となる．なお，単純に $\boldsymbol{x}_1$ を上位に，$\boldsymbol{x}_2$ を下位に配置した場合，$p_k = k\,(k = 1, 2, \ldots, K)$，$p'_l = l + K\,(l = 1, 2, \ldots, L)$ となる．従って，$\boldsymbol{x}_1$ および $\boldsymbol{x}_2$ の選択方法は $\begin{pmatrix} N \\ K \end{pmatrix} = \begin{pmatrix} N \\ L \end{pmatrix}$ 種類存在する．P を変更した場合のベクトル $\boldsymbol{x} = {}^t(\boldsymbol{x}_1, \boldsymbol{x}_2)$ に対する線形変換は，式 (1) における変換行列 $\boldsymbol{A}$ の列ベクトルを入替える事により，実現される．同様に，変換後，符号化すべきベクトル $\boldsymbol{y}_1$ と $\boldsymbol{y}_2 = \boldsymbol{0}$ の $\boldsymbol{y}$ における位置も変更可能であり，$\boldsymbol{y}_1$ と $\boldsymbol{y}_2$ 各成分の添字の集合をそれぞれ $Q = \{q_1, q_2, \ldots, q_K\}$，$Q^c = \{q'_1, q'_2, \ldots, q'_L\}$ とすると，Q の変更は，変換行列 $\boldsymbol{A}$ の行ベクトルを入替える事に相当するため，式 (1) の線形変換は次式の様に変更される．

$$\begin{aligned} &{}^t\!\left(y_{q_1}\, y_{q_2} \cdots y_{q_K}\, y_{q'_1}\, y_{q'_2} \cdots y_{q'_L}\right) \\ &= \tilde{\boldsymbol{A}}\; {}^t\!\left(x_{p_1}\, x_{p_2} \cdots x_{p_K}\, x_{p'_1}\, x_{p'_2} \cdots x_{p'_L}\right) \end{aligned} \tag{5}$$

ただし，

$$\begin{aligned} \tilde{\boldsymbol{A}} &= \left(\begin{array}{c|c} \tilde{\boldsymbol{A}}_{1\,1} & \tilde{\boldsymbol{A}}_{1\,2} \\ \hline \tilde{\boldsymbol{A}}_{2\,1} & \tilde{\boldsymbol{A}}_{2\,2} \end{array}\right) \\ &= \left(\begin{array}{ccc|ccc} a_{q_1\,p_1} & \cdots & a_{q_1\,p_K} & a_{q_1\,p'_1} & \cdots & a_{q_1\,p'_L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{q_K\,p_1} & \cdots & a_{q_K\,p_K} & a_{q_K\,p'_1} & \cdots & a_{q_K\,p'_L} \\ \hline a_{q'_1\,p_1} & \cdots & a_{q'_1\,p_K} & a_{q'_1\,p'_1} & \cdots & a_{q'_1\,p'_L} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{q'_L\,p_1} & \cdots & a_{q'_L\,p_K} & a_{q'_L\,p'_1} & \cdots & a_{q'_L\,p'_L} \end{array}\right) \end{aligned} \tag{6}$$

3　H.264/AVC への適用

H.264/AVC の整数精度 DCT は次に示す 4 次の直交行列 $\boldsymbol{C} = (c_{m\,n})\,(m, n = 1, 2, 3, 4)$ によって定義される [5]．

$$\boldsymbol{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix} \tag{7}$$

4×4 画素の画像信号 $\boldsymbol{F} = (f_{m\,n})$ に対する線形変換 $\boldsymbol{G} = (g_{m\,n})$ は次式で表される．

$$\boldsymbol{G} = \boldsymbol{C}\,\boldsymbol{F}\,{}^t\boldsymbol{C} \tag{8}$$

前節の手法を画像信号に適用するために，$\boldsymbol{F}$ と $\boldsymbol{G}$ を $4^2 = 16$ 次元のベクトルで表現する．$\boldsymbol{F}$ および $\boldsymbol{G}$ をラスタスキャン順でベクトル化した場合，変換行列 $\boldsymbol{A}$ は次式で表される．

$$\boldsymbol{A} = \boldsymbol{C} \otimes \boldsymbol{C} \tag{9}$$

ここで，$\boldsymbol{C} \otimes \boldsymbol{C}$ は 2 つの $\boldsymbol{C}$ の Kronecker 積であり，$\boldsymbol{A}$ は 16 次の正方行列になる [6]．式 (9) によって変換行列が示されたため，Shen らの手法に倣い，最適な領域内画素の配置場所及びパディング行列を求める．

まず，一般に変換係数 $\boldsymbol{y}$ はジグザグスキャン順に符号化されるため，$\boldsymbol{y}_1$ の各成分の添字 q_k は

$$\{1,2,5,9,6,3,4,7,10,13,14,11,8,12,15,16\} \tag{10}$$

の最初から K 個を採用し，残りを q'_l に用いればよい．

一方，4×4 画素ブロック内において領域内とすべき画素位置，つまり P の決定は，符号化すべき変換係数 $\boldsymbol{y}_1$ が

$$\boldsymbol{y}_1 = \boldsymbol{Q}\cdot \boldsymbol{x}_1 \quad (\boldsymbol{Q} = \tilde{\boldsymbol{A}}_{1\,1} - \tilde{\boldsymbol{A}}_{1\,2}\tilde{\boldsymbol{A}}_{2\,2}^{-1}\tilde{\boldsymbol{A}}_{2\,1}) \tag{11}$$

によって定まる事を用いる．つまり，

$$||\boldsymbol{y}_1||^2 \le ||\boldsymbol{Q}||^2 \cdot ||\boldsymbol{x}_1||^2 \tag{12}$$

の関係より，$\boldsymbol{Q}$ のノルムが最小となる p_k および p'_l を採用する．

4　計算結果

全ての K に対する計算結果として，ラスタスキャンでベクトル化した場合における，領域内画素の成分の添字集合 P（$K \le 8$）と領域外画素のそれ P^c（$K > 8$），およびパディング行列 $\boldsymbol{\Pi}$（ただし，$K > 8$ の時は ${}^t\boldsymbol{\Pi}$）を以下に示す．$\boldsymbol{\Pi}$ の第 l 行は p'_l に，第 k 列は p_k に対応し，添字集合 P および P^c はラスタスキャン順の小さいものから並ぶものとする．$\boldsymbol{Q}$ のノルムが等しい組合せが複数生じた場合は，一種のみを記した．

$K=1, L=15$ の場合

$$P = \{15\} \tag{13}$$

全ての成分が 1　(14)

$K=2, L=14$ の場合

$$P = \{4,9\} \tag{15}$$

$$\begin{pmatrix} 0.000 & 1.000 \\ 0.250 & 0.750 \\ 0.750 & 0.250 \\ 0.000 & 1.000 \\ 0.250 & 0.750 \\ 0.750 & 0.250 \\ 1.000 & 0.000 \\ 0.250 & 0.750 \\ 0.750 & 0.250 \\ 1.000 & 0.000 \\ 0.000 & 1.000 \\ 0.250 & 0.750 \\ 0.750 & 0.250 \\ 1.000 & 1.000 \end{pmatrix} \tag{16}$$

$K=3, L=13$ の場合

$$P = \{2,12,13\} \tag{17}$$

$$\begin{pmatrix} 1.067 & -0.267 & 0.200 \\ 0.867 & 0.533 & -0.400 \\ 0.800 & 0.800 & -0.600 \\ 0.800 & -0.200 & 0.400 \\ 0.733 & 0.067 & 0.200 \\ 0.600 & 0.600 & -0.200 \\ 0.533 & 0.867 & -0.400 \\ 0.267 & -0.067 & 0.800 \\ 0.200 & 0.200 & 0.600 \\ 0.067 & 0.733 & 0.200 \\ -0.067 & 0.267 & 0.800 \\ -0.200 & 0.800 & 0.400 \\ -0.267 & 1.067 & 0.200 \end{pmatrix} \tag{18}$$

$K=4, L=12$ の場合

$$P = \{4,5,12,13\} \tag{19}$$

$$\begin{pmatrix} 0.667 & 0.667 & -0.667 & 0.333 \\ 0.750 & 0.500 & -0.500 & 0.250 \\ 0.917 & 0.167 & -0.167 & 0.083 \\ 0.083 & 0.833 & 0.167 & -0.083 \\ 0.250 & 0.500 & 0.500 & -0.250 \\ 0.333 & 0.333 & 0.667 & -0.333 \\ -0.333 & 0.667 & 0.333 & 0.333 \\ -0.250 & 0.500 & 0.500 & 0.250 \\ -0.083 & 0.167 & 0.833 & 0.083 \\ 0.083 & -0.167 & 0.167 & 0.917 \\ 0.250 & -0.500 & 0.500 & 0.750 \\ 0.333 & -0.667 & 0.667 & 0.667 \end{pmatrix} \tag{20}$$

$K=5, L=11$ の場合

$$P = \{1,4,6,13,16\} \tag{21}$$

$$\begin{pmatrix} 0.750 & 0.250 & 0.000 & 0.000 & 0.000 \\ 0.250 & 0.750 & 0.000 & 0.000 & 0.000 \\ 0.188 & -0.188 & 1.000 & 0.063 & -0.063 \\ -0.375 & 0.375 & 1.000 & -0.125 & 0.125 \\ -0.563 & 0.562 & 1.000 & -0.188 & 0.188 \\ -0.313 & -0.188 & 1.000 & 0.562 & -0.063 \\ -0.375 & -0.125 & 1.000 & 0.375 & 0.125 \\ -0.500 & 0.000 & 1.000 & 0.000 & 0.500 \\ -0.562 & 0.062 & 1.000 & -0.187 & 0.687 \\ 0.000 & 0.000 & 0.000 & 0.750 & 0.250 \\ 0.000 & 0.000 & 0.000 & 0.250 & 0.750 \end{pmatrix} \tag{22}$$

$K=6, L=10$ の場合

$$P = \{1,4,6,12,13,15\} \tag{23}$$

$$\begin{pmatrix} 0.457 & 0.283 & 0.522 & -0.522 & -0.217 & 0.478 \\ -0.043 & 0.783 & 0.522 & -0.522 & -0.217 & 0.478 \\ 0.457 & -0.217 & 0.522 & 0.478 & 0.283 & -0.522 \\ -0.326 & 0.370 & 0.913 & 0.087 & -0.130 & 0.087 \\ -0.196 & 0.522 & 0.348 & 0.652 & 0.022 & -0.348 \\ -0.043 & -0.217 & 0.522 & 0.478 & 0.783 & -0.522 \\ -0.326 & -0.130 & 0.913 & 0.087 & 0.370 & 0.087 \\ -0.304 & -0.022 & 0.652 & 0.348 & -0.022 & 0.348 \\ -0.196 & 0.022 & 0.348 & -0.348 & 0.522 & 0.652 \\ 0.391 & -0.043 & -0.696 & 0.696 & -0.043 & 0.696 \end{pmatrix} \tag{24}$$

$K=7, L=9$ の場合

$$P=\{1,4,6,7,9,14,16\} \tag{25}$$

$$\begin{pmatrix} 0.571 & 0.143 & 0.571 & 0.000 & -0.571 & 0.429 & -0.143 \\ 0.476 & 0.286 & -0.524 & 1.000 & -0.476 & 0.524 & -0.286 \\ 0.381 & -0.071 & 0.381 & 0.000 & 0.619 & -0.381 & 0.071 \\ -0.429 & 0.643 & 0.571 & 0.000 & 0.429 & -0.571 & 0.357 \\ -0.286 & -0.071 & 0.714 & 0.000 & 0.286 & 0.286 & 0.071 \\ -0.095 & -0.357 & -0.095 & 1.000 & 0.095 & 0.095 & 0.357 \\ -0.429 & 0.143 & 0.571 & 0.000 & 0.429 & -0.571 & 0.857 \\ 0.238 & 0.143 & -0.762 & 0.000 & 0.762 & 0.762 & -0.143 \\ 0.286 & -0.429 & -0.714 & 1.000 & -0.286 & 0.714 & 0.429 \end{pmatrix} \tag{26}$$

$K=L=8$ の場合

$$P=\{1,3,4,6,11,13,14,16\} \tag{27}$$

$$\begin{pmatrix} 0.250 & 0.667 & -0.250 & 0.667 & -0.667 & -0.250 & 0.333 & 0.250 \\ 0.562 & -0.500 & 0.187 & 0.500 & 0.500 & 0.437 & -0.500 & -0.188 \\ -0.125 & 0.333 & 0.125 & 0.333 & 0.667 & 0.125 & -0.333 & -0.125 \\ -0.188 & -0.500 & 0.938 & 0.500 & 0.500 & 0.188 & -0.500 & 0.062 \\ 0.062 & -0.500 & 0.188 & 0.500 & 0.500 & 0.938 & -0.500 & -0.188 \\ -0.125 & -0.333 & 0.125 & 0.667 & 0.333 & 0.125 & 0.333 & -0.125 \\ -0.188 & -0.500 & 0.438 & 0.500 & 0.500 & 0.187 & -0.500 & 0.562 \\ 0.250 & 0.333 & -0.250 & -0.667 & 0.667 & -0.250 & 0.667 & 0.250 \end{pmatrix} \tag{28}$$

$K=9, L=7$ の場合

$$P^c=\{3,5,6,10,11,12,15\} \tag{29}$$

$${}^t\begin{pmatrix} -0.062 & 0.500 & -0.188 & -0.188 & 0.250 & 0.000 & 0.437 \\ 0.250 & 0.000 & 0.750 & 0.250 & -0.500 & 0.000 & -0.750 \\ 0.688 & 0.000 & -0.187 & -0.188 & -0.250 & -0.500 & 0.187 \\ 1.000 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 & 1.000 \\ -0.750 & 0.000 & 0.250 & 0.250 & 0.000 & 1.000 & -0.750 \\ -0.250 & 1.000 & 0.750 & 0.750 & 0.000 & 0.000 & -0.250 \\ 0.312 & -0.500 & -0.562 & -0.562 & -0.250 & 0.000 & -0.188 \\ -0.250 & 0.000 & 0.250 & 0.750 & 0.500 & 0.000 & 0.750 \\ 0.063 & 0.000 & -0.062 & -0.063 & 0.250 & 0.500 & 0.562 \end{pmatrix} \tag{30}$$

$K=10, L=6$ の場合

$$P^c=\{3,6,9,11,12,15\} \tag{31}$$

$${}^t\begin{pmatrix} 0.063 & -0.562 & 0.062 & 0.812 & 0.562 & 0.562 \\ 0.250 & 0.750 & -0.250 & -0.750 & -0.250 & -0.750 \\ 0.688 & -0.188 & 0.188 & -0.062 & -0.312 & 0.188 \\ -0.250 & 0.750 & 0.250 & -0.750 & -0.750 & -0.250 \\ 1.000 & 0.000 & 0.000 & 1.000 & 0.000 & 1.000 \\ -0.750 & 0.250 & -0.250 & -0.250 & 0.750 & -0.750 \\ 0.000 & 0.000 & 1.000 & 1.000 & 1.000 & 0.000 \\ 0.188 & -0.187 & 0.688 & -0.062 & 0.188 & -0.312 \\ -0.250 & 0.250 & -0.750 & -0.250 & -0.750 & 0.750 \\ 0.063 & -0.062 & 0.062 & 0.312 & 0.562 & 0.562 \end{pmatrix} \tag{32}$$

$K = 11, L = 5$ の場合

$$P^c = \{3, 6, 10, 11, 15\} \tag{33}$$

$$^t\begin{pmatrix} 0.062 & -0.562 & -0.188 & 0.375 & 0.562 \\ 0.250 & 0.750 & 0.250 & -0.500 & -0.750 \\ 0.688 & -0.188 & -0.062 & 0.125 & 0.188 \\ -0.250 & 0.750 & 0.000 & -0.250 & -0.250 \\ 1.000 & 0.000 & 0.000 & 1.000 & 1.000 \\ -0.750 & 0.250 & 0.000 & -0.750 & -0.750 \\ 0.000 & 0.000 & 0.750 & 0.250 & 0.000 \\ 0.000 & 0.000 & 0.250 & 0.750 & 0.000 \\ 0.187 & -0.188 & -0.562 & -0.375 & -0.312 \\ -0.250 & 0.250 & 0.750 & 0.500 & 0.750 \\ 0.063 & -0.062 & -0.188 & -0.125 & 0.562 \end{pmatrix} \tag{34}$$

$K = 12, L = 4$ の場合

$$P^c = \{2, 7, 10, 15\} \tag{35}$$

$$^t\begin{pmatrix} 0.583 & -0.083 & 0.083 & -0.083 \\ 0.667 & 0.333 & -0.333 & 0.333 \\ -0.250 & -0.250 & 0.250 & -0.250 \\ -0.500 & 0.000 & -0.500 & 0.500 \\ 0.667 & 0.333 & 0.667 & -0.667 \\ -0.167 & 0.667 & -0.167 & 0.167 \\ 0.167 & -0.167 & 0.667 & -0.167 \\ -0.667 & 0.667 & 0.333 & 0.667 \\ 0.500 & -0.500 & 0.000 & -0.500 \\ -0.250 & 0.250 & -0.250 & -0.250 \\ 0.333 & -0.333 & 0.333 & 0.667 \\ -0.083 & 0.083 & -0.083 & 0.583 \end{pmatrix} \tag{36}$$

$K = 13, L = 3$ の場合

$$P^c = \{7, 10, 11\} \tag{37}$$

$$^t\begin{pmatrix} 0.375 & 0.375 & 0.500 \\ -0.750 & -0.500 & -0.750 \\ 0.750 & 0.000 & 0.250 \\ -0.375 & 0.125 & 0.000 \\ -0.500 & -0.750 & -0.750 \\ 1.000 & 1.000 & 1.000 \\ 0.500 & -0.250 & -0.250 \\ 0.000 & 0.750 & 0.250 \\ 0.000 & 0.250 & 0.750 \\ 0.125 & -0.375 & 0.000 \\ -0.250 & 0.500 & -0.250 \\ 0.250 & 0.000 & 0.750 \\ -0.125 & -0.125 & -0.500 \end{pmatrix} \tag{38}$$

$K = 14, L = 2$ の場合

$$P^c = \{10, 11\} \tag{39}$$

$$^t\begin{pmatrix} 0.375 & 0.125 \\ -0.500 & 0.000 \\ 0.000 & -0.500 \\ 0.125 & 0.375 \\ -0.750 & -0.250 \\ 1.000 & 0.000 \\ 0.000 & 1.000 \\ -0.250 & -0.750 \\ 0.750 & 0.250 \\ 0.250 & 0.750 \\ -0.375 & -0.125 \\ 0.500 & 0.000 \\ 0.000 & 0.500 \\ -0.125 & -0.375 \end{pmatrix} \tag{40}$$

$K = 15, L = 1$ の場合

$$P^c = \{11\} \tag{41}$$

$$^t\begin{pmatrix} -0.250 \\ 0.500 \\ -0.500 \\ 0.250 \\ 0.500 \\ -1.000 \\ 1.000 \\ -0.500 \\ -0.500 \\ 1.000 \\ 0.500 \\ 0.250 \\ -0.500 \\ 0.500 \\ -0.250 \end{pmatrix} \tag{42}$$

以上から分かるように，$\mathbf{\Pi}$ の各行ベクトル成分の総和は 1 になる．証明は [3] に記してある．

5 むすび

本論文では，H.264/AVC で用いられる直交変換行列に対するパディング行列の導出を行った．ここで求めたパディング行列を用いた場合，領域外画素と同数の変換係数が 0 となり，符号化すべき変換係数がジグザグスキャンの順に並ぶとの利点を有する．導出の際，H.264/AVC は変換行列は既存の国際標準のそれと比べ次数が低いため，領域内として判断すべき画素の最適な位置を全探索により求めることが出来た．今後の課題として，符号化への実装が挙げられる．

参考文献

[1] W. Li, J. Ohm, M. Schaar, H. Jiang, and S. Li, “MPEG-4 video verification model version 18.0,” ISO/IEC JTC1/SC29/WG11, N3908, Jan. 2001.

[2] G. Shen and M. Liou, “Arbitrarily shaped Transform coding based on a new padding technique,” IEEE Trans. Image Processing, vol.11, no.1, Jan. 2001.

[3] 黒木祥光，上繁義史，“任意形状 DCT 符号化における新たなパディング法，” 画像符号化シンポジウム (PCSJ)2002，pp.31-32，Nov. 2002.

[4] 大久保　榮監修，“H.264/AVC 教科書，” インプレス，2004.

[5] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, “Low-complexity transform and quantization in H.264/AVC,” IEEE Trans. Cir. and Sys. for Video Tech., vol. 13, no. 7, July 2003.

[6] 原島　博監修，“映像情報符号化，” オーム社，2001.