Proceedings of 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, October 4-7, 2009

Arbitrarily Shaped Transform Coding Based on Modification of Pixels in Shapes

Yuta Higuchi* and Yoshimitsu Kuroki*
* Kurume National College of Technology
1-1-1, Komorino, Kurume-shi, Fukuoka 838-8555 Japan
Tel: +81-942-35-9392 Fax: +81-942-35-9398
E-mail: kuroki@kurume-nct.ac.jp

***Abstract*—Arbitrarily shaped transform coding is a tool to achieve object-based coding. Two types of arbitrarily shaped coding, which guarantee equivalence between the number of pixels inside shapes and DCT coefficients to be coded, are proposed as [3] and [4]. These methods obtain same DCT coefficients, and the difference is that the former calculates outside pixel values from the inside whereas the latter varies the inside pixels by themselves. They are based on 1D-DCT; therefore, 1D-DCT must be executed twice, namely horizontal and vertical directions. This procedure does not guarantee the equivalence. In this paper, we extend their method to two-dimensional transform. In addition, we discuss methods to optimize position of DCT coefficients and pixels using statistical model. We embedded the proposed method in H.264/AVC, and simulation results indicate all proposed method performed almost same in coding efficiency.**

***Index Terms*—Arbitrarily shaped transform coding, padding techniques, H.264/AVC**

I. INTRODUCTION

Arbitrarily shaped transform coding is a tool to achieve object-based coding such as SA-DCT [1] and padding technique. The padding technique called low-pass extrapolation (LPE) [2] is used for the MPEG-4 standards. LPE generates rectangular blocks to fill pixels of outside shape by filtering boundary pixels, and transformed coefficients concentrate on low frequencies. However, this method doesn't guarantee that the coefficients of high frequency are always set to zero. As one of the solutions, Shen, etc. and Koga, etc. propose different techniques respectively which guarantee equivalence between the number of pixels inside shapes and DCT coefficients to be coded [3] [4]. These methods obtain same DCT coefficients, but calculation process of the coefficients is different. The former calculate outside pixel values from the inside, whereas the latter varies the inside pixels by themselves. Their methods must be executed twice for images, horizontal and vertical direction, because they are based on 1D-DCT. Then, this procedure does not guarantee the equivalence. Moreover, a position of coefficients is not optimized though their methods optimize a position of pixels in 1D-DCT. These problems seem to decrease the coding efficiency.

In this paper, we extend their method to two-dimensional transform used in H.264/AVC, and also introduce the technique to optimize position of DCT coefficients and pixels taking acount of the scan of DCT coefficients on encoding. This method guarantees the equivalence and is able to optimize a position of coefficients and pixels.

II. CONVENTIONAL METHODS

A. The technique proposed by Shen, etc. [3]

In $\boldsymbol{R}^N$ space, we assume that a vector in a shape and a vector out of a shape are denoted as $\boldsymbol{x} = (x_0, x_1, \cdots, x_{k-1})^T$ and $\boldsymbol{y} = (y_0, y_1, \cdots, y_{N-k-1})^T$. Then, the base-$N$ transform by $\boldsymbol{U}$ operated on the vector $(\boldsymbol{x}^T \boldsymbol{y}^T)^T$ can be written as follows:

$$\begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} = \boldsymbol{U} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}_{00} & \boldsymbol{U}_{01} \\ \boldsymbol{U}_{10} & \boldsymbol{U}_{11} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}, \quad (1)$$

where $\boldsymbol{U}_{00}$, $\boldsymbol{U}_{01}$, $\boldsymbol{U}_{10}$ and $\boldsymbol{U}_{11}$ are submatrices of order $k \times k$, $k \times (N-k)$, $(N-k) \times k$ and $(N-k) \times (N-k)$. In (1), we can vary elements of $\boldsymbol{Y}$ voluntary; therefore, setting $\boldsymbol{Y} = 0$ will result in the solution of the equivalence between the number of pixels and coefficients. In that case, when $\boldsymbol{U}_{11}$ is a regular matrix, $\boldsymbol{y}$ is shown as follows:

$$\boldsymbol{y} = -\boldsymbol{U}_{11}^{-1} \boldsymbol{U}_{10} \boldsymbol{x}. \quad (2)$$

In addition, based on (1) and (2), we can derive

$$\boldsymbol{X} = (\boldsymbol{U}_{00} - \boldsymbol{U}_{01} \boldsymbol{U}_{11}^{-1} \boldsymbol{U}_{10}) \boldsymbol{x} = \boldsymbol{K} \boldsymbol{x}. \quad (3)$$

Now, the elements of $\boldsymbol{x}$ can be arranged freely with the unit base of $\boldsymbol{R}^N$ space. The operation which arranges the elements of $\boldsymbol{x}$ is equal to the exchange of the column vectors in $\boldsymbol{U}$, and the number of the arrangements is expressed as ${}_N C_k$. It is defined that the arrangement to minimize $||K||$ is the most optimal arrangements.

B. The technique proposed by Koga, etc. [4]

[4] proposes another technique which sets values of $\boldsymbol{y}$ to zero and varies the elements of $\boldsymbol{x}$ themselves in (1). In $\boldsymbol{R}^N$ space, this method varies a vector in a shape $\boldsymbol{x}^{(N)} = (x_0 \boldsymbol{e}_0^{(N)} + x_1 \boldsymbol{e}_1^{(N)} + \cdots + x_{k-1} \boldsymbol{e}_{k-1}^{(N)})$ of dimension N to $\boldsymbol{x}^{(N)} + \Delta \boldsymbol{x}^{(N)}$ using the variabled vector $\Delta \boldsymbol{x}^{(N)} = (\Delta x_0 \boldsymbol{e}_0^{(N)} + \Delta x_1 \boldsymbol{e}_1^{(N)} + \cdots + \Delta x_{k-1} \boldsymbol{e}_{k-1}^{(N)})$ of dimension N. In $\boldsymbol{x}^{(N)}$ and $\Delta \boldsymbol{x}^{(N)}$, $\boldsymbol{e}_i^{(N)}$ means the i-th unit based in $\boldsymbol{R}^{(N)}$ spase. Then, a projection of $\boldsymbol{x} + \Delta \boldsymbol{x}$ toward the liner space $L[\boldsymbol{u}_{\boldsymbol{0}}, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_{k-1}]$ satisfies

$$\boldsymbol{u}_i \cdot \left\{ (x^{(N)} + \Delta \boldsymbol{x}^{(N)}) - \sum_{j=0}^{k-1} X_j \boldsymbol{u}_j \right\} = 0 \quad (4)$$

$$(i = 0, 1, \cdots k-1),$$

where $\boldsymbol{u}_i$ is the ith row vector in $\boldsymbol{U}$. In addition, a projection of $\boldsymbol{X}^{(N)}$ toward unit based, namely inverse transform, must be equal to $\boldsymbol{x}^{(N)}$, and it follows that

$$\left(\sum_{j=0}^{k-1} X_j \boldsymbol{u}_j \right) \cdot \boldsymbol{e}_i = x_i \quad (i = 0, 1, \cdots, k-1). \quad (5)$$

From (4) and (5), we obtain

$$\boldsymbol{X}^{(k)} = (\boldsymbol{U}_{00}^T)^{-1} \cdot \boldsymbol{x}^{(k)}, \quad (6)$$

$$\boldsymbol{U}_{00} \cdot \Delta \boldsymbol{x}^{(k)} = \boldsymbol{X}^{(k)} - \boldsymbol{U}_{00} \cdot \boldsymbol{x}^{(k)}. \quad (7)$$

In these equations, $\boldsymbol{x}^{(k)}$ and $\Delta \boldsymbol{x}^{(k)}$ are dimension k, and $\boldsymbol{U}_{00}$ is equal to the submatrix defined in (1). Based on (6) and (7), we can drive

$$\Delta \boldsymbol{x}^{(k)} = (\boldsymbol{U}_{00}^{-1} (\boldsymbol{U}_{00}^{-1})^T - \boldsymbol{I}) \cdot \boldsymbol{x}^{(k)} = \boldsymbol{A} \cdot \boldsymbol{x}^{(k)}. \quad (8)$$

This method also consider optimal arrangement of pixels. In $\boldsymbol{x}$ and $\Delta \boldsymbol{x}$, we can choose the unit bases to express freely. It is defined that the arrangement to minimize $||(\boldsymbol{U}_{00}^T)^{-1}||$ is the most optimal in arrangements.

C. Extension to 2D-transform

Their methods are based on 1D-DCT; therefore, they must be executed twice for images, horizontal and vertical directions. This operation does not guarantee a equivalence between the number of inside pixels and DCT coefficients. To this problem, Kuroki, etc. propose a solution which extends [3] to two-dimensional transform [5]. This method can also be applied to [4].

We suppose that a image block and a DCT coefficient block are denoted as $\boldsymbol{F} = (f_{ij})$ $(i, j = 0, 1, \cdots, N-1)$ and $\boldsymbol{G} = (g_{kl})$ $(k, l = 0, 1, \cdots, N-1)$. Then, base-$N$ transform by $\boldsymbol{U}$ is expressed as

$$\boldsymbol{G} = \boldsymbol{U} \boldsymbol{F} \boldsymbol{U}^T. \quad (9)$$

Equation (9) is also seen that $\boldsymbol{G}$ is expressed as twice operations on $\boldsymbol{U}$. Now, let new vectors to arrante elements of $\boldsymbol{F}$ and $\boldsymbol{G}$ in raster scan order are denoted as $\boldsymbol{f} = (f_{00}, f_{01}, \cdots, f_{(N-1)(N-1)})^T$ and $\boldsymbol{g} = (g_{00}, g_{01}, \cdots, g_{(N-1)(N-1)})^T$, the transform of (9) is re-written as

$$\boldsymbol{g} = \acute{\boldsymbol{U}} \boldsymbol{f}. \quad (10)$$

In (10), $\acute{\boldsymbol{U}}$ is the kronecker of $\boldsymbol{U}$ and itself ($\acute{\boldsymbol{U}} = \boldsymbol{U} \otimes \boldsymbol{U}$).

III. Arrangement of pixels and DCT coefficients

In this paper, we employ the orthogonal transform used in H.264/AVC [6], which is shown as follows:

$$\boldsymbol{U} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}. \quad (11)$$

As mentioned above, [3] and [4] are based on orthonormal transform; therefore, it is necessary to normalize (11). In addition, these proposals do not optimize a position of DCT coefficients though they optimize a position of pixels. In this section, we introduce two types of technique to optimize a position of DCT coefficients and pixels taking acuount of scan of DCT coefficients on encoding. These proposals are expected to improve coding efficiency.

A. Arrangement 1

In [3] and [4], the DCT coefficients appear in the raster scan order, however the encode is carried out in the zigzag order. Therefore, the coding efficiency becomes worse, because values of zero are read on the scan of encoding. As a solution for the problem, we produce a new arrangement method to appear the DCT coefficients in zigzag order. This operation is equal to exchange the row vectors in transform matrix of $\acute{\boldsymbol{U}}$ [as defined in (10)]; therefore, this method optimizes inside pixel positions using the transform matrix after the DCT coefficients position are optimized.

B. Arrangement 2

We also introduce the other optimum arrangement method using statistical model. Arrangement 1 fixes the position of DCT coefficients in zigzag order, whereas this arrangement selects a optimum position in all combination on DCT coefficients and inside pixels.

In (10), we assume that a vector in a shape and a vector of transformed coefficients are denoted as $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{X}}$ of dimention k. Then, the transform $\tilde{\boldsymbol{x}}$ into $\tilde{\boldsymbol{X}}$ is expressed as

$$\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{U}} \tilde{\boldsymbol{x}}, \quad (12)$$

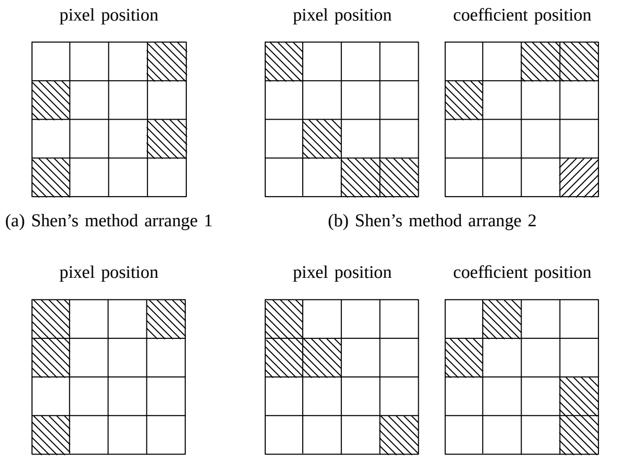
$\tilde{\boldsymbol{U}}$ is varied depending on each arbitrary coding method and selected positions of inside pixels and DCT coefficients. From (12), we can derive

$$\tilde{\boldsymbol{X}} \otimes \tilde{\boldsymbol{X}} = \tilde{\boldsymbol{U}} \tilde{\boldsymbol{x}} \otimes \tilde{\boldsymbol{x}} \tilde{\boldsymbol{U}}^T. \quad (13)$$

Now, the correlation model of pixels is expressed as follows:

$$\boldsymbol{R} = E[\boldsymbol{x} \otimes \boldsymbol{x}] = \begin{pmatrix} 1 & \rho^1 & \cdots & \rho^{k-1} \\ \rho^1 & 1 & \cdots & \rho^{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{k-1} & \rho^{k-2} & \cdots & 1 \end{pmatrix}. \quad (14)$$

In (14), factorials of ρ vary with the unit based in $\boldsymbol{x}$. Based on (13) and (14), we can get



(a) Shen's method arrange 1 (b) Shen's method arrange 2

(c) Koga's method arrange 1 (d) Koga's method arrange 2

Fig. 1. The positions of each methods when the number of inside pixels is 4

$$E[\tilde{\boldsymbol{X}} \otimes \tilde{\boldsymbol{X}}] = \tilde{\boldsymbol{U}}\boldsymbol{R}\tilde{\boldsymbol{U}}^T. \tag{15}$$

This method is extended to 2D transform; therefore, the total number of the arrangements is expressed as ${}_{N^2}C_k$. Accordingly, we define that the arrangement to minimize a sum of all diagonal elements of $E[\tilde{\boldsymbol{X}} \otimes \tilde{\boldsymbol{X}}]$ is the most optimal arrangement. At this case, we set a value of ρ to 0.9 tentatively.

C. Difference of position in each method

As for these optimum methods, the positions of inside pixels and DCT coefficients is different respectively, as shown in Fig.1 which showes the positions of each coding methods using two types of proposed arrangement method, when the number of inside pixels is 4. The blocks shown in slash in Fig.1 are the positions of inside pixels and DCT coefficients, and in the method of [3], the white blocks are varied actually, because [3] calculate outside pixel values from the inside. A glance at Fig.1 will reveal that different optimized positions in same valuation two or more exists in each methods. It follows form these results that the pixel varying matrix is different depending on each methods and arrangement, which mean that the coding efficiency is different, too.

IV. EXPERIMENTAL RESULTS

We extend [3] and [4] to two-dimentional transform, and encode a arbitrarily shaped image using the two types of the position arrangement method. In this paper, we also embed the propose method in H.264/AVC. To accommodate the proposed methods in H.264/AVC standards, we employ the reference software var.14.0 produced by JVT [7]. This standard encodes a difference between the origin image and forecast image, that operation called intra-prediction. What has to be noticed is arbitrarily shaped images contain outside pixels. Therefore, we adjust the flow to estimae the difference beyond a shape at zero.

To confirm the utility of that, we derive the Rate-PSNR, and compare it in each method, as shown in Fig.2,3,4. The arbitrarily shaped images that we use are "*Akiyo*", "*News*" and

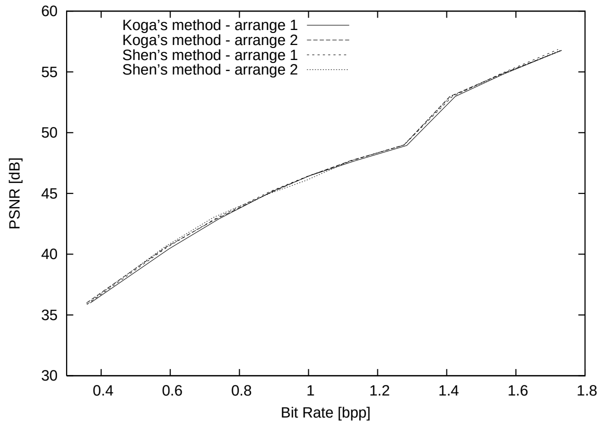


Fig. 2. Rate-PSNR performance of "*Akiyo*"

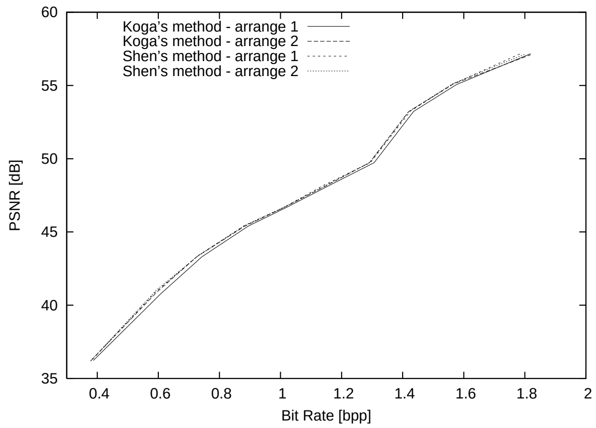


Fig. 3. Rate-PSNR performance of "*News*"

"*Weather*" of size 176 $\times$ 144, and we exchange a value of QP (quantization Parameter) from 10 to 34 in this encoding to get more results with a different PSNR rate. The coding conditions in this experiment are as follows.

- Fix the block size in intra-prediction and coding operation
- Not use the inter-prediction

We see from the figures that all proposed method performed almost same in coding efficiency. In addition, we see that the arrangements reach the high coding efficiency in high rate encoding.

Incidentally, we set a value of ρ as 0.9 tentatively in the proposed position arrangement using statistical model. However, we can choose the different values of ρ; therefore, It is considerable that the position of inside pixels and DCT coefficients change to the other position in setting different values of ρ.

V. CONCLUSION

In this paper, we extend arbitrarily shaped transform codings proposed [3] and [4] to two-dimensional transform, and introduce two types of arrangement method to optimize the position of inside pixels and DCT coefficients to improve the

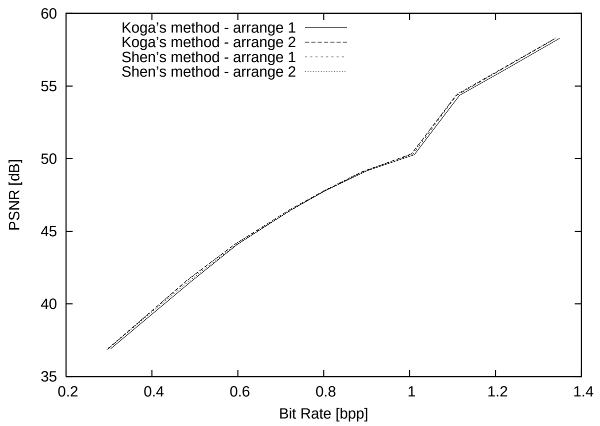


Fig. 4. Rate-PSNR performance of ”*Weather*”

coding efficiency. The results indicate all proposed method performed almost same in coding efficiency. In addition, we see that the arrangements reach the high coding efficiency in high rate encoding.

The future direction of this study will be using the other value of ρ to get the different positions, and correspondence with Inter-prediction and other brock size in coding operation.

REFERENCES

[1] T.Sikora and B.Makai、”Shape-Adaptive DCT for Generic Coding of Video”, IEEE Trans. Circuit and System for Video Technology, Vol. 5, No. 1, pp.59-62, Feb.1995.

[2] W.Li, J.R.Ohm, M.van der Shaar, H.Jiang, and S.Li ”MPEG-4 video verification model version 18.0”, ISO/IEC JTC1/SC29/WG11, N3908, Jan. 2001.

[3] G.shen, B.Zeng, and M.L.Liou ”Arbitrarily shaped transform coding baced on a new padding technique”, IEEE Trans. Circuit and Systems for Video Technology, vol.11, no.1, Jan. 2001.

[4] Yuichiro Koga and Yoshimitsu Kuroki, ”Arbitrarily Shaped Transform Coding Based On Correctiion Of Pixels In Shapes”, Information and Systems Society Special Issue, 2008 IEICE General Conference, pp.91, March 2008 (Japanese),

[5] Yoshimitsu Kuroki and Yoshifumi Uesige, ”A padding technique for orthogonal transform on H.264/AVC”, IPSJ SIG Technical Reports 2005-AVM-50, pp.17-22, Oct.2005 (Japanese),

[6] H.Malvar, A.Hallapuro, M.Karczewicz, and L.Kerofsky ”Low-Complexity Transform and Quantization with 16-bit Arithmetic for H26L”, IEEE International Conference on Image Processing, pp.489-492, March 2008

[7] Joint Video Team (JVT), ”Reference Software ver.14.0”, http://iphome.hhi.de/suehring/tml