

# Transform-Domain Intra Prediction for H.264

Chen Chen and Ping-Hao Wu

Graduate Institute of Communications Engineering,  
National Taiwan University,  
Taipei, Taiwan 10617, R.O.C.  
Email: {r92942081, b89901043}@ntu.edu.tw

Homer Chen

Department of Electrical Engineering, GICE, and INM,  
National Taiwan University,  
Taipei, Taiwan 10617, R.O.C.  
Email: homer@cc.ee.ntu.edu.tw

**Abstract**—H.264/AVC is the newest video coding standard jointly developed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. In contrast to some previous coding standards such as H.263+ and MPEG-4 Part-2, where intra prediction is performed in the transform domain, the intra prediction of H.264 is completely defined in the pixel domain. This presents a challenge to multimedia systems in which transcoding is conducted in the transform domain for the purpose of computational efficiency. In this paper, we show how to obtain the transform domain predictions for various intra modes of H.264. We begin by converting the intra prediction from the pixel domain to the transform domain through matrix manipulation. Then we show how the operations involved in the matrix manipulation can be simplified. A computational complexity analysis of each intra prediction mode of H.264 is provided.

## I. INTRODUCTION

In view of the successful adoption of MPEG-2 technology in the DVD and DTV industries and the growing market potential of the newly developed H.264 standard, we are motivated to investigate the conversion of existing MPEG-2 multimedia content to the H.264 format. The overall architecture of our proposed scheme for MPEG-2 to H.264 transcoding is described in [6]. In this paper, we present the details of our transform-domain approach for H.264 intra prediction.

H.264/AVC [1] is the newest video coding standard jointly developed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. It provides better compression efficiency than previous video coding standards. Transcoding is one of the key technologies for enabling interoperability between different systems and devices [1], [2]. The purpose of transcoding is to convert a multimedia signal from one format to another. The format conversion is characterized by bitrate, frame rate, coding standard, etc.

In contrast to some previous coding standards such as H.263+ and MPEG-4 Part-2, where intra prediction is performed in the transform domain, the intra prediction of H.264 is completely defined in the pixel domain by referring to neighboring samples of the previously coded blocks. Transcoding in the transform (or compressed) domain has been a desired approach [3], [4] because it avoids complete decoding and re-encoding, which is

computationally expensive. Therefore, there is a need to investigate transform-domain intra prediction for H.264.

This paper is organized as follows. In Section II, we describe the proposed transform-domain intra prediction algorithm. The complexity of the algorithm is presented in Section III. Finally, a conclusion is given in Section IV.

## II. INTRA PREDICTION

In H.264, there are a total of 9 prediction modes for a  $4 \times 4$  sub-block and 4 for a  $16 \times 16$  macroblock. If a sub-block or a macroblock is to be coded in intra mode, a prediction block is formed based on the neighboring samples of previously-coded blocks that are to the left and/or immediately above the block to be coded. The prediction block is then subtracted from the current block prior to encoding. In this section, the transform-domain intra prediction for luma samples is discussed. The transform-domain intra prediction of chroma samples can be easily derived in a similar way and is not described here.

### A. Intra $4 \times 4$

The intra prediction of  $4 \times 4$  blocks defined by H.264 is illustrated in Fig. 1(a), where the symbols a to p denote the pixels of current block, and the symbols A to M denote the neighboring pixels based on which the prediction block is calculated. The direction of prediction is shown in Fig. 1(b). Due to the page limit, we choose the first four modes of the  $4 \times 4$  intra prediction to illustrate our approach.

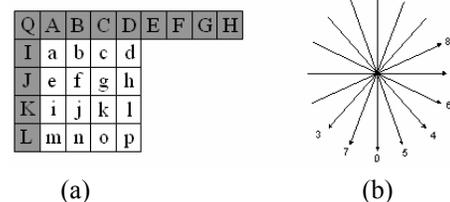


Figure 1. Illustration of  $4 \times 4$  intra prediction

### 1) Vertical

The prediction samples in this mode (mode 0) are obtained from the four neighboring pixels (A to D) above the block to be coded. Sample A is copied to every pixel in the first column of the block, sample B is copied to every pixel in the second column of the block, and so on.

This work was supported in part by the National Science Council of Taiwan under contract NSC93-2752-E-002-006-PAE.

Mathematically, we can express this prediction block by the following matrix equation:

$$\begin{bmatrix} A & B & C & D \\ A & B & C & D \\ A & B & C & D \\ A & B & C & D \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}, \quad (1)$$

where the symbol “ $\times$ ” represents “don’t care” pixel. Denoting the modified DCT used in H.264 by  $T(\cdot)$  and applying it both sides of (1), we have:

$$\begin{aligned} & T \begin{pmatrix} A & B & C & D \\ A & B & C & D \\ A & B & C & D \\ A & B & C & D \end{pmatrix} \\ &= T \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} T \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{pmatrix} \\ &= \begin{bmatrix} 1 & -1.2649 & 1 & -0.6325 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{pmatrix}. \quad (2) \end{aligned}$$

Therefore, this  $4 \times 4$  intra prediction mode can be performed in the transformed domain. Furthermore, given the transform coefficients of the upper block, a total of 8 multiplications and 12 additions are required to compute the prediction samples of the current block in the transform domain.

### 2) Horizontal

Following the procedure described above, the prediction block of this mode can be expressed as:

$$\begin{bmatrix} I & I & I & I \\ J & J & J & J \\ K & K & K & K \\ L & L & L & L \end{bmatrix} = \begin{bmatrix} \times & \times & \times & I \\ \times & \times & \times & J \\ \times & \times & \times & K \\ \times & \times & \times & L \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad (3)$$

which has the same computational complexity as (2).

### 3) DC

In this mode, the prediction samples are obtained from both upper and left blocks and can be expressed as:

$$\begin{aligned} & \left( \frac{A+B+C+D+I+J+K+L}{8} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \left( \frac{A+B+C+D}{8} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + \left( \frac{I+J+K+L}{8} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= \frac{1}{8} \begin{bmatrix} A & B & C & D \\ A & B & C & D \\ A & B & C & D \\ A & B & C & D \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} + \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} I & I & I & I \\ J & J & J & J \\ K & K & K & K \\ L & L & L & L \end{bmatrix}. \quad (4) \end{aligned}$$

The modified DCT of every matrix in (4), except the one with all elements equal to one, have already been computed in the previous two modes. Note that

$$T \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5)$$

Therefore, only one multiplication and one addition are required to obtain the prediction samples of this mode in the transform domain.

### 4) Diagonal\_Down\_Left

The operation involved in this mode is much more complicated than the previous three modes.

First, we divide the prediction samples of this mode into two terms according to the sources of the pixel value (from the upper block or from the upper-right block):

$$\begin{aligned} & \frac{1}{4} \begin{bmatrix} A+2B+C & B+2C+D & C+2D+E & D+2E+F \\ B+2C+D & C+2D+E & D+2E+F & E+2F+G \\ C+2D+E & D+2E+F & E+2F+G & F+2G+H \\ D+2E+F & E+2F+G & F+2G+H & G+3H \end{bmatrix} \\ &= \frac{1}{4} \begin{pmatrix} A+2B+C & B+2C+D & C+2D & D \\ B+2C+D & C+2D & D & 0 \\ C+2D & D & 0 & 0 \\ D & 0 & 0 & 0 \end{pmatrix} \\ &+ \begin{pmatrix} 0 & 0 & E & 2E+F \\ 0 & E & 2E+F & E+2F+G \\ E & 2E+F & E+2F+G & F+2G+H \\ 2E+F & E+2F+G & F+2G+H & G+3H \end{pmatrix}. \quad (6) \end{aligned}$$

Then each row in the first term can be calculated as follows:

$$\begin{aligned} & \begin{bmatrix} A+2B+C & B+2C+D & C+2D & D \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{bmatrix}, \quad (7) \end{aligned}$$

$$\begin{aligned} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ B+2C+D & C+2D & D & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \end{bmatrix}, \quad (8) \end{aligned}$$

$$\begin{aligned} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C+2D & D & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}, \quad (9) \end{aligned}$$

and

$$\begin{aligned} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ A & B & C & D \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (10)$$

The summation of the four blocks described by (7)-(10) gives rise to the first matrix on the right-hand side of (6). The second matrix on the right-hand side of (6) can be calculated in a similar way.

At the first sight, it may seem that this intra prediction mode involves a lot of matrix multiplications. A closer look at the above four equations indicates that the first matrices on the right-hand side of (7)-(10) have a vertical shift relationship. We exploit such relationship to simplify the computation.

First, take the modified transform of the first matrices described above and denote them by  $\mathbf{U}_1$ ,  $\mathbf{U}_2$ ,  $\mathbf{U}_3$ , and  $\mathbf{U}_4$ , respectively:

$$\begin{aligned} \mathbf{U}_1 &= \mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{U}_2 = \mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ \mathbf{U}_3 &= \mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{U}_4 = \mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Then, multiply  $\mathbf{U}_1$  with the modified DCT of the upper block and denote the resulting matrix by  $\mathbf{Y}_1$ . We thus have:

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{U}_1 \mathbf{X} \\ &= \begin{bmatrix} a & -b & a & c \\ 1.264a & -1.264b & 1.264a & 1.264c \\ a & -b & a & c \\ 0.632a & -0.632b & 0.632a & 0.632c \end{bmatrix} \mathbf{X}, \end{aligned} \quad (11)$$

where  $a=0.25$ ,  $b=0.316$ ,  $c=0.159$ , and

$$\mathbf{X} = \mathbf{T} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ A & B & C & D \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix}. \quad (12)$$

Let

$$\begin{aligned} S_0 &= a(X_{11} + X_{31}) - bX_{21} + cX_{41}, \\ S_1 &= a(X_{12} + X_{32}) - bX_{22} + cX_{42}, \\ S_2 &= a(X_{13} + X_{33}) - bX_{23} + cX_{43}, \\ S_3 &= a(X_{14} + X_{34}) - bX_{24} + cX_{44}. \end{aligned} \quad (13)$$

Then, (11) can be rewritten as:

$$\mathbf{Y}_1 = \begin{bmatrix} S_0 & S_1 & S_2 & S_3 \\ 1.264S_0 & 1.264S_1 & 1.264S_2 & 1.264S_3 \\ S_0 & S_1 & S_2 & S_3 \\ 0.632S_0 & 0.632S_1 & 0.632S_2 & 0.632S_3 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \\ \tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_3 \end{bmatrix}, \quad (14)$$

where  $\tilde{\mathbf{Y}}_i, 1 \leq i \leq 3$ , denote different row vectors of  $\mathbf{Y}_1$ . Note that the computation of the matrix  $\mathbf{Y}_1$  is now reduced to 20 multiplications and 12 additions. Because of the vertical shift relationship, the remaining three matrices  $\mathbf{Y}_2 = \mathbf{U}_2 \mathbf{X}$ ,  $\mathbf{Y}_3 = \mathbf{U}_3 \mathbf{X}$ , and  $\mathbf{Y}_4 = \mathbf{U}_4 \mathbf{X}$  are also composed of  $\tilde{\mathbf{Y}}_1$ ,  $\tilde{\mathbf{Y}}_2$ , and  $\tilde{\mathbf{Y}}_3$ .

$$\mathbf{Y}_2 = \mathbf{U}_2 \mathbf{X} = \begin{bmatrix} \tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_3 \\ -\tilde{\mathbf{Y}}_1 \\ -\tilde{\mathbf{Y}}_2 \end{bmatrix}, \quad \mathbf{Y}_3 = \mathbf{U}_3 \mathbf{X} = \begin{bmatrix} \tilde{\mathbf{Y}}_1 \\ -\tilde{\mathbf{Y}}_3 \\ -\tilde{\mathbf{Y}}_1 \\ \tilde{\mathbf{Y}}_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{Y}_4 = \mathbf{U}_4 \mathbf{X} = \begin{bmatrix} \tilde{\mathbf{Y}}_1 \\ -\tilde{\mathbf{Y}}_2 \\ \tilde{\mathbf{Y}}_1 \\ -\tilde{\mathbf{Y}}_3 \end{bmatrix}. \quad (15)$$

Therefore,  $\mathbf{Y}_2$ ,  $\mathbf{Y}_3$ , and  $\mathbf{Y}_4$  are readily available without any additional calculation. Moreover, since the matrices  $\mathbf{Y}_1$ ,  $\mathbf{Y}_2$ ,  $\mathbf{Y}_3$ , and  $\mathbf{Y}_4$  are needed in the prediction process of other modes, we can store them in memory to avoid unnecessary computation.

Using a similar procedure described previously, the prediction processes of the remaining intra 4x4 modes can be derived easily. Due to the page limit, we omit the derivation of other modes here.

### B. Intra 16x16

Intra 16x16 prediction of H.264 contains four modes: *Vertical*, *Horizontal*, *DC* and *Plane* mode. The transform domain prediction of the first three modes can be derived by the procedure described previously in Section II.A. Here, we focus on the *Plane* mode which is more complicated than the other three modes. Fig. 2 shows the notations used to denote neighboring pixels around the current block for which a prediction block in the transform domain is to be determined.

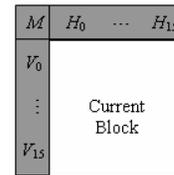


Figure 2. Illustration of the 16x16 intra prediction.

The prediction samples of the *Plane* mode can be represented as:

$$a \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} + b \begin{bmatrix} -7 & -6 & \dots & 8 \\ \vdots & \vdots & \vdots & \vdots \\ -7 & -6 & \dots & 8 \end{bmatrix} + c \begin{bmatrix} -7 & \dots & -7 \\ -6 & \dots & -6 \\ \vdots & \dots & \vdots \\ 8 & \dots & 8 \end{bmatrix}, \quad (16)$$

where

$$\begin{aligned}
a &= 16(H_{15} + V_{15}) \\
b &= (5H + 32) \gg 6 \\
c &= (5V + 32) \gg 6
\end{aligned} \tag{17}$$

and

$$H = 8(H_{15} - M) + \sum_{h=0}^6 (h+1)(H_{8+h} - H_{6-h}), \tag{18}$$

$$V = 8(V_{15} - M) + \sum_{v=0}^6 (v+1)(V_{8+v} - V_{6-v}). \tag{19}$$

The values  $a$ ,  $b$ , and  $c$  are dependent on the neighboring pixel values. We now show how to obtain  $H$ . The calculation of  $V$ ,  $H_{15}$ , and  $V_{15}$  can be easily accomplished by following the same procedure.

Since we only have  $4 \times 4$  blocks of modified DCT coefficients, we expand (18) into five terms according to the pixel location:

$$\begin{aligned}
H &= (5H_{12} + 6H_{13} + 7H_{14} + 8H_{15}) \\
&\quad + (H_8 + 2H_9 + 3H_{10} + 4H_{11}) \\
&\quad - (3H_4 + 2H_5 + H_6) - (7H_0 + 6H_1 + 5H_2 + 4H_3) - 8M, \tag{20}
\end{aligned}$$

where  $(5H_{12} + 6H_{13} + 7H_{14} + 8H_{15})$  can be determined by making use of the following equation:

$$\begin{aligned}
&\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ H_{12} & H_{13} & H_{14} & H_{15} \end{bmatrix} \begin{bmatrix} 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 \\ 7 & 7 & 7 & 7 \\ 8 & 8 & 8 & 8 \end{bmatrix} \\
&= (5H_{12} + 6H_{13} + 7H_{14} + 8H_{15}) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{21}
\end{aligned}$$

According to (5), the value of  $(5H_{12} + 6H_{13} + 7H_{14} + 8H_{15})$  can be determined by extracting the DC value of the modified transform of the left hand side of (21). The remaining terms of  $H$  in (20) can be obtained in a similar way.

### III. COMPUTATIONAL COMPLEXITY

In this section, we show the computational complexity of the proposed algorithm by the number of operations required to obtain the prediction samples.

First, note that the prediction blocks are computed row by row and the pre-computed matrices are shared by different modes. For example, the values  $S_0$ ,  $S_1$ ,  $S_2$ , and  $S_3$  computed in the *Diagonal\_Down\_Left* mode can also be used in the prediction process of the *Vertical\_Left*, the *Vertical\_Right*, the *Diagonal\_Down\_Right*, and the *Horizontal\_Down* modes. Thus we only need to calculate those matrices once.

The number of operations required for each intra prediction mode is shown in Table I and Table II. Note that in the first two modes of the  $16 \times 16$  intra prediction, the

TABLE I. COMPUTATIONAL COMPLEXITY OF  $4 \times 4$  INTRA PREDICTION

Mode	Multiplication	Addition
Vertical	8	12
Horizontal	8	12
DC	1	1
Diagonal_Down_Left	168	136
Diagonal_Down_Right	232	200
Vertical_Right	192	176
Horizontal_Down	192	176
Vertical_Left	128	112
Horizontal_Up	64	48

TABLE II. COMPUTATIONAL COMPLEXITY OF  $16 \times 16$  INTRA PREDICTION

Mode	Multiplication	Addition
Vertical	0	0
Horizontal	0	0
DC	1	7
Plane	26	72

operation counts are all zero because the data required by these modes have already been calculated in the first two modes of intra  $4 \times 4$  prediction.

### IV. CONCLUSION

In this paper, we have provided a sound theoretical basis for converting various intra prediction modes of H.264 from the pixel domain to the transform domain. We have also analyzed the operations involved in the conversion and shown means for efficient implementation of the algorithm. This work is motivated by a practical need for conducting transcoding in the transform domain. The methods developed are readily applicable to multimedia gateways or systems for converting, for example, MPEG-2 videos to the H.264 format.

### REFERENCES

- [1] A. Vetro, C. Christopoulos, and Huifang Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Processing Magazine*, pp. 18-29, Mar. 2003.
- [2] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," to appear in Proc. IEEE, Dec. 2004.
- [3] P. A. A. Assuncao and M. Ghanbari, "Transcoding of MPEG-2 video in the frequency domain," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing, ICASSP'97*, Munich, Germany, Apr. 1997, vol. 4, pp. 2633-2636.
- [4] T. Shanableh, and M. Ghanbari, "Hybrid DCT/pixel domain architecture for heterogeneous video transcoding," *Signal Processing: Image Commun.*, vol. 18, no. 8, pp. 601-620, Sep. 2003.
- [5] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 2003.
- [6] C. Chen, P.-H. Wu, and H. Chen, "MPEG-2 to H.264 transcoding," *Picture Coding Symposium*, Dec. 2004.