Adaptive Zero-Coefficient Distribution Scan for Inter Block Mode Coding of H.264/AVC

Jing-Xin WANG†, Nonmember and Alvin W.Y. SU†, Member

SUMMARY Scanning quantized transform coefficients is an important tool for video coding. For example, the MPEG-4 video coder adopts three different scans to get better coding efficiency. This paper proposes an adaptive zero-coefficient distribution scan in inter block coding. The proposed method attempts to improve H.264/AVC zero coefficient coding by modifying the scan operation. Since the zero-coefficient distribution is changed by the proposed scan method, new VLC tables for syntax elements used in context-adaptive variable length coding (CAVLC) are also provided. The savings in bit-rate range from 2.2% to 5.1% in the high bit-rate cases, depending on different test sequences.

key words: H.264/AVC, adaptive zero-coefficient distribution scan, inter block coding

1. Introduction

H.264/AVC [1], [2] adopts many tools such as intra prediction, variable block-size motion estimation and in-loop de-blocking filters to get more coding efficiency. Among most of the tools, a scan operation is used to transfer 2-D transform coefficients to 1-D ones. Then, coders such as context-adaptive variable length coding (CAVLC), which uses several context-based variable length coding (VLC) tables, can be applied. Although the use of several scan methods had been implemented in MPEG-4 [3], H.264/AVC only adopts the traditional zigzag scan. In [4]–[6], vertical scan and horizontal scan are utilized in vertical and horizontal prediction modes in intra block coding, respectively. Choi [7] reported six fixed scan methods for all intra prediction modes of each 4×4 block. Tao [8] presented more number of fixed scan methods. The adaptive scan pattern in intra block coding was proposed in [9]. Comparably, there has been less discussion in the research on scan methods in inter block coding. In Sun’s method [10], there are four adaptive scan methods in inter block coding when AVS [11] coder is considered. The scan methods are decided by the sorted absolute values of quantized transform coefficients in the previous frame. Sun’s method can be directly applied because AVS uses 8×8 blocks, but H.264/AVC baseline profile only uses 4×4 blocks.

In our paper, coding efficiencies associated with four different scan methods that are zigzag scan, sorted coefficients scan, sorted non-zero coefficients scan and sorted zero coefficients scan for CAVLC in H.264/AVC are first discussed for 4×4 block. Then, three scan strategies are proposed for adaptive zero-coefficient distribution scan (AZDS). Since multiple scan methods are employed, the side information must be used to indicate which scan method is used for each macroblock (MB). More side information bits are required if more scan methods are used. Therefore, coding performance of using different numbers of scan methods is discussed. Since the zero-coefficient distribution is changed by AZDS, new VLC tables for syntax elements of CAVLC are also provided.

The rest of this paper is organized as follows. Section 2 introduces the CAVLC and coding efficiencies of the four basic scans mentioned above are discussed. Section 3 describes the proposed AZDS and the new VLC tables. Section 4 shows the simulation results, and Sect. 5 gives our conclusion.

2. H.264/AVC CAVLC

There are five syntax elements defined in H.264/AVC CAVLC.

Coeff_Token: the total number of non-zero coefficients (TotalCoeffs) and the number trailing ±1 (T1).
T1_Sign: sign bit for each T1.
Level: the values of non-zero coefficients except T1s.
TotalZero: the total number of zero coefficients before the last non-zero coefficient.
Run_Before: the number of consecutive preceding zeros for each non-zero coefficient.

These five syntax elements are used for coding two groups of coefficients: zero coefficients and non-zero coefficients. TotalZero and Run_Before are for the zero coefficients and the other three elements are for the non-zero coefficients. Different scan methods may produce different sets of syntax elements and the bit amount is also affected. First, in addition to zigzag scan, three other scans are implemented in order to show their effects. They are called sorted coefficients (SC) scan, sorted non-zero coefficients (SNC) scan and sorted zero coefficients (SZC) scan, respectively. SC uses absolute values of all coefficients to sort the coefficients, SNC uses absolute value of all non-zero coefficients to sort the coefficients and SZC moves all zero coefficients to the trail of non-zero coefficients such that TotalZero is zero. An example is shown in Fig. 1. ZeroLeft is the number
of preceding zero coefficients for each non-zero coefficient. Figure 1 (a) shows the quantized coefficients and the scan results obtained with four different scan methods. The corresponding variable length code and syntax elements using the original H.264/AVC coder are shown in Fig. 1 (b)–(e). It can be seen that SC is the most efficient one. To find the bit amount associated with the above four scan methods, Akiyo, Forman and Stefan in CIF resolution are encoded. The number of reference frames is 5, I-Frame period is 300 and rate-distortion optimization (RDO) is enabled. In order to compare their performances, identical coefficients to be encoded for each scan method are necessary. Therefore, zigzag scan is used to calculate the bit-rate for RDO algorithm when determining the best block mode. Then 2-D coefficients are scanned with different scan methods and encoded. The simulation results are shown in Table 1. The saving in bit-rate is calculated against the zigzag scan. To get the maximum improvement in coding efficiency of each scan method, the side information of each scan method is skipped in the simulation results. SC is combining two methods of SZC and SNC. As expected, SC has the best performance, but the most performance of SC is obtained from SZC. It is also found that changing the scan method is quite effective in higher bit-rate (lower QP) cases. Table 2 shows the percentage of the number of bits encoded in non-zero coefficients, zero coefficients and header. Header includes motion vectors, coding block mode and coded block pattern (CBP). The average TotalZero of a block in QP = 1, 8 and 16 are 0.16, 0.51 and 1.6, respectively. Although the average TotalZero in a block is few in higher bit-rate, the percentage of the number of bits encoded zero coefficients is more than 21.95%. Therefore, several Run_Befores are used to indicate where the zero coefficients locate in a block and the most number of bits encoded zero coefficients is spent to encode the Run_Befores. The performance of SZC is mainly gained from reducing the number of Run_Befores such as Fig. 1 (e). SZC can reduce the number of bits in higher bit-rate. Run_Before is the more important parameter for SC's and SZC's good performance. Therefore, we propose the zero-coefficient distribution scan to approach the strategy of decreasing the number of Run_Befores.

### Table 1: The average savings in bit-rate for three sequences.\(^1\)

| Methods | SC | SZC | SNC |
|---------|----|-----|-----|
| 1       | 19.86% | 14.31% | 5.54% |
| 8       | 21.63% | 17.81% | 3.81% |
| 16      | 19.79% | 17.18% | 2.59% |
| 32      | 8.26% | 7.45% | 0.8%  |
| 48      | 2.71% | 2.63% | 0.06% |

### Table 2: The percentage of the number of bits encoded three parameters.\(^1\)

| QP | Non-zero coefficients | Zero coefficients | Header |
|----|------------------------|-------------------|--------|
| 1  | 73.89%                 | 21.95%            | 4.16%  |
| 8  | 61.68%                 | 31.26%            | 6.4%   |
| 16 | 44.62%                 | 31.26%            | 24.12% |
| 32 | 27.93%                 | 15.73%            | 56.34% |
| 48 | 7.87%                  | 3.36%             | 88.77% |

#### 3. Adaptive Zero-Coefficient Distribution Scan

Based on Sect. 2, Run_Before is the more important parameter for SC's and SZC's good performance. Thus, three new scan strategies based on the distribution of zero coefficients of 4×4 blocks are built for zero coefficients coding. Before describing the new scan methods, the indicator of zero coefficients in a 4×4 block is defined as

\[
\text{Zero}_{M,b}^c(i) = \begin{cases} 
1, & \text{if } QDCT_{M,b}^F(i) = 0 \\
0, & \text{otherwise}
\end{cases},
\]

for \(i = 0, 1, \ldots, 15\). \(^{15}\)

where \(b\) is the block index, \(M\) is the MB index, \(F\) is the frame index, and \(i\) is the index of the \(i\)-th coefficient of a 4×4 block. \(QDCT_{M,b}^F(i)\) is the \(i\)-th quantized coefficient for the \(b\)-th 4×4 block of the \(M\)-th MB of the \(F\)-th frame. Because the proposed method is used only in inter mode, \(\text{Zero}_{M,b}^c(i)\) are set to zero for all \(i\) and \(b\) if a MB is coded in intra mode.

### 3.1 Zero-Coefficient Distribution Scans

Before describing the new scan methods, three functions related to zero-coefficient distribution are applied to obtain the zero-coefficient distribution scans. First, we define \(\text{NumMBZero}_{M,F}(i)\) in Eq. (2) to estimate the number of accumulated zeros at the \(i\)-th location in the \(M\)-th MB.

\[
\text{NumMBZero}_{M,F}(i) = a * \text{NumMBZero}_{M,F-1}(i) + \sum_{b=0}^{15} \text{Zero}_{M,b}^c(i), \text{for } i = 0, 1, \ldots, 15,
\]

where \(a\) represents a forgetting factor. It is used
to reduce the effect from previous frame. Second, $\text{NumFrameZero}_F(i)$ defined in Eq. (3) represents the total number of accumulated zeros of all sub-blocks at the $i$-th location in the $F$-th frame.

$$\text{NumFrameZero}_F(i) = \sum_{all \ M} \sum_{b=0}^{15} \text{Zero}_{M,b}^F(i),$$

for $i = 0, 1, \ldots, 15$,  

(3)

The best result of the adaptive scan method is performed by that each block has its own scan order such as SC, but the huge number of bits to encode the side information is expected. To decrease the side information and gain more number of scan order, MBs of a frame can be partitioned into several clusters whose scan orders are similar. In each cluster, the total number of accumulated zeros is calculated by

$$\text{NumClusterZero}_F,C(i) = \sum_{all\ m_c} \sum_{b=0}^{15} \text{Zero}_{m_c,b}^F(i),$$

for $i = 0, 1, \ldots, 15$,  

(4)

where $m_c = \{m : 0 \leq m < TM$ and if the $m$-th MB $\in$ the $C$-th Cluster$, TM$ is the total number of MBs of a frame and $C$ is the cluster index. By using Eq. (2), MBZero scan is built based on $\text{NumMBZero}_{M,F}(i)$. Similarly, FrameZero scan is built based on $\text{NumFrameZero}_F(i)$ and ClusterZero scan is built based on $\text{NumClusterZero}_{F,C}(i)$. The scan orders of three scan methods are obtained from the sorted results of the correspondent zero-coefficient distribution functions defined by Eq. (2), Eq. (3) and Eq. (4), respectively. An example of FrameZero scan is shown in Fig. 2. The detail ClusterZero scan method will be described later.

Fig. 2 An example of FrameZero scan: (a) $\text{NumFrameZero}_F(i)$, for $i = 0, 1, 2, \ldots, 15$ (b) scan order of FrameZero scan of $(F+1)$-th frame.

The coefficients of the $M$-th MB of the $(F+1)$-th frame is scanned based on $\text{NumMBZero}_{M,F}(i)$, $\text{NumFrameZero}_F(i)$ and $\text{NumClusterZero}_{F,C}(i)$, for $i = 0, 1, \ldots, 15$ and all clusters, respectively. There can be only one scan method for all $4 \times 4$ blocks in a MB. Which scan method is applied is determined by whether it produces the least number of bits to encode a MB among all methods. Then the corresponding side information has to be encoded, too. No side information is necessary when MB is coded in skip mode and intra mode or when coded block pattern (cbp) of MB is 0. Moreover, if the $F$-th frame is an I-frame and the $(F+1)$-th frame is encoded as a P-frame, zigzag scan will be used for the $(F+1)$-th frame exclusively.

Fig. 3 The encoding flow using the ClusterZero scan in a GOP.
Finally, Fig. 3 is used to explain the encoding flow using the ClusterZero scan. For an I-frame, calculation of Eq. (4) is skipped. When the first P-frame of a GOP is encountered, the clusters are initialized as follows. In this P-frame, MB with its index falling in the interval \( [\frac{NC}{NC + TM}, \frac{NC}{NC + 1} - TM] \) is forced to belong to the \( C \)-th cluster, where \( NC \) is the number of clusters. Let the first P-frame in a GOP be the \( F \)-th frame. Then, \( NumClusterZero_{F,C}(i) \) in Eq. (4), for all \( i \) and \( C \), are calculated. The scan orders of all clusters of the \((F+1)\)-th frame are refreshed based on \( NumClusterZero_{F,C}(i) \), respectively. In the \((F+1)\)-th frame, the scan order of the \( C \)-th cluster is applied to the \( M \)-th MB if this specific scan order produces the least number of bits to encode the MB. Moreover, after the MB is encoded using the ClusterZero scan, \( NumClusterZero_{F+1,C}(i) \) updated from \( Zero_{F+1,M} \) for all \( i \) and \( b \). However, if either MBZero scan or FrameZero scan is applied instead of ClusterZero scan, \( NumClusterZero_{F+1,C}(i) \) for all \( C \) are not updated. Finally, the scan orders of all clusters for the \((F+2)\)-th frame are refreshed from \( NumClusterZero_{F+1,C}(i) \), respectively.

### 3.2 Adaptive Zero-Coefficient Distribution Scan

This subsection describes how to select a target scan method. It was found that the simulation results using rearranged coefficients will be closer to SZC when more scan methods can be selected, but the number of side information bits also increases. To find a good balance, experiments listed in Table 3 are performed. Akiyo, Foreman and Football in CIF resolution are encoded. In Table 1, the improvement of coding efficiency has a high relation with QP. TotalCoeffs depend on QP and are accessible in the decoder. \( SSTC \) (Sum of Square TotalCoeffs) that is defined in Eq. (5) is used in AZDS.

\[
SSTC = \sum_{b=0}^{15} \text{TotalCoeffs}(b)^2
\]

where \( \text{TotalCoeffs}(b) \) is the total number of non-zero coefficients of \( b \)-th block in a MB. The average numbers of bits saved in a MB are shown in Fig. 4 when more than one scan methods are available. In Fig. 4, the side information bits have been considered to give the true total number of bits saved in the encoding process. It can be seen that more bits are saved when \( SSTC \) becomes larger. From Fig. 4(a), zigzag scan is enough if \( SSTC < 8 \). Similarly, ZigZag scan and FrameZero scan can be used if \( 8 \leq SSTC < 16 \). Table 4 shows the number of scan methods to be applied for different \( SSTCs \). The value of NumClusterZero of ClusterZero (NC=2) and ClusterZero (NC=5) are calculated independently to decide their own scan order, respectively. Figure 5(a) and (b) shows new encoding and decoding flows using AZDS. In the encoder, \( SSTC \) has to be calculated first in order to select the number of available scan methods. The scan method consuming the least number of bits to encode a MB is applied. When writing bitstream, Coeff_Token of all 4×4 blocks in a MB are encoded first. Then the other 4 syntax elements of each 4×4 block and the side information are encoded. Therefore, Coeff_Token including TotalCoeffs can be decoded first to calculate \( SSTC \) in the decoder. Finally, the scan orders of three proposed scans of the \((F+1)\)-th frame are refreshed based on their corresponding zero-coefficient distribution functions of the \( F \)-th frame defined in Eq. (2), Eq. (3) and Eq. (4), respectively.

### 3.3 The New VLC Table for Run_Before

When AZDS is applied, Run_Before is affected. Moreover, the number of preceding non-zero coefficients for each non-zero coefficient, denoted by NonZeroLeft, also affects Run_Before. Table 5 shows the percentages of Run_Before for different values of NonZeroLeft when ZeroLeft is 4. The testing sequences are Akiyo, Foreman, and Football in CIF resolution for \( QP = 8, 12, \ldots, 48 \). Run_Before = 4 happens more frequently than Run_Before = 0 when NonZeroLeft = 1. The new VLC tables are required if AZDS is applied. For example, the new VLC tables and the original VLC table when ZeroLeft is 4 are shown in Table 6. It can be seen that the original VLC table does not fully match the situations given in Table 5 when NonZeroLeft = 1 and NonZeroLeft \( \geq 3 \). The new VLC tables of whole ZeroLefts are provided in Appendix.

---

**Table 3** Required side information bits when using different numbers of scan methods.

| Number of bits spent for side information in a MB | Number of available scan methods | Available scan methods |
|-------------------------------------------------|----------------------------------|------------------------|
| 0                                               | 1                                | ZigZag                 |
| 1                                               | 2                                | ZigZag, FrameZero      |
| 2                                               | 4                                | ZigZag, MBZero, ClusterZero (NC=2) |
| 3                                               | 8                                | ZigZag, MBZero, FrameZero, ClusterZero (NC=5) |

**Table 4** Using \( SSTC \) to determine the number of available scan methods.

| \( SSTC \) | Number of available scan methods to be selected |
|------------|-----------------------------------------------|
| \( < 8 \)  | 1                                             |
| \( 8 \leq SSTC < 16 \) | 2                                         |
| \( 16 \leq SSTC < 40 \) | 4                                         |
| \( 40 \leq SSTC \) | 8                                         |
The seven sequences used in this paper are Foreman (CIF, 300 frames), Akiyo (CIF, 300 frames), Mobile (CIF, 300 frames), Coastguard (CIF, 300 frames), Table tennis (352×240, 112 frames), Garden (CIF, 250 frames) and Parkrun (1280×720, 600 frames) in 30 fps. The JM 14.0 encoder[12] serves as our reference. The parameters of H.264/AVC encoder are set as those shown in Table 7. The RDO is enabled and only first frame is I frame. The forgetting factor mentioned in Eq. (2) is 0.9 that is obtained from experimentation. The improvements brought by the proposed method for all seven test sequences when compared to the standard method are shown in Table 8. The percentage of side information bits is also shown. Since
AZDS is only performed in inter mode coding, the coding results of I frames are excluded. The savings in bit-rate are over 4% for Garden when QPs are from 16 to 32. 5.12% bit-rate reduction is achieved in the Table tennis when QP is 16. The improvement decreases when the QP increases. This can be seen in Akiyo because there is little information to be coded when QP is high and the side information brought by the proposed method becomes too large when compared to the improvement in texture coding. The percentages of improvement in coding efficiency using four scan methods are shown in Table 9. ClusterZero (NC=5) is most effective in four scan methods. But it includes five scan orders, the average improvement of each scan order in ClusterZero is close to that of FrameZero and MBZero. FrameZero, MBZero and ClusterZero (NC=2) gain more efficiency when QP becomes large. The average increase of computational complexity of AZDS is shown in Table 10. The maximum average increase of computational complexity are 6.45% in QP = 8. NumFrameZero and NumMBZero must be calculated in all MBs unless that the encoded mode of the MB is skip mode. NumClusterZero is only calculated in that MBs adopt the cluster scan method. In the Table 9, the number of MBs that use cluster scan is decreasing and the number of MBs that are skip mode is increasing in large QP. Therefore, the increase of computational complexity is reduced when QP becomes larger.

Table 8  The savings in bit-rate for seven sequences.

| Methods         | JM14.0 (Zigzag only) | AZDS without new VLC tables | AZDS with new VLC tables |
|------------------|-----------------------|-----------------------------|--------------------------|
| Sequence         | QP        | PSNR  (dB) | bit-rate (Kbps) | PSNR  (dB) | Total saving in bit-rate | PSNR  (dB) | Total saving in bit-rate | Side information |
| Forman (CIF)     | 8         | 53.59    | 9758.23         | 53.59    | 2.23%                 | 53.59    | 3.89%                 | 0.31%     |
|                  | 16        | 46.31    | 3843.83         | 46.31    | 3.04%                 | 46.32    | 3.95%                 | 0.78%     |
|                  | 24        | 39.62    | 923.65          | 39.62    | 1.96%                 | 39.62    | 2.26%                 | 1.30%     |
|                  | 32        | 34.04    | 239.48          | 34.04    | 0.19%                 | 34.04    | 0.08%                 | 2.79%     |
| Akiyo (CIF)      | 8         | 53.72    | 2735.83         | 53.72    | 1.31%                 | 53.72    | 2.29%                 | 0.53%     |
|                  | 16        | 47.2     | 670.64          | 47.2     | 1.30%                 | 47.2     | 1.53%                 | 1.21%     |
|                  | 24        | 42.43    | 158.65          | 42.43    | 0.37%                 | 42.43    | 0.34%                 | 1.58%     |
|                  | 32        | 37.17    | 41.72           | 37.17    | −0.17%                | 37.17    | −0.20%                | 0.68%     |
| Mobile (CIF)     | 8         | 53.6     | 13325.13        | 53.6     | 1.37%                 | 53.6     | 2.62%                 | 0.26%     |
|                  | 16        | 46.52    | 7361.67         | 46.52    | 1.88%                 | 46.53    | 3.16%                 | 0.47%     |
|                  | 24        | 38.66    | 3036.48         | 38.66    | 1.79%                 | 38.67    | 2.43%                 | 0.97%     |
|                  | 32        | 31.25    | 837.4           | 31.25    | 0.72%                 | 31.26    | 1.10%                 | 1.41%     |
| Coastguard (CIF)| 8         | 53.64    | 11154.56        | 53.64    | 1.78%                 | 53.64    | 2.69%                 | 0.31%     |
|                  | 16        | 46.38    | 5741.34         | 46.38    | 2.55%                 | 46.38    | 3.00%                 | 0.60%     |
|                  | 24        | 38.72    | 2161.46         | 38.72    | 2.53%                 | 38.72    | 2.57%                 | 1.17%     |
|                  | 32        | 32.12    | 583.75          | 32.12    | 1.31%                 | 32.11    | 1.12%                 | 1.60%     |
| Table tennis     | 8         | 53.71    | 10746.45        | 53.71    | 2.76%                 | 53.71    | 4.85%                 | 0.25%     |
| (352×240)        | 16        | 45.45    | 4950.68         | 45.45    | 3.33%                 | 45.47    | 5.12%                 | 0.47%     |
|                  | 24        | 37.92    | 1505.75         | 37.92    | 2.68%                 | 37.92    | 3.73%                 | 0.92%     |
|                  | 32        | 31.67    | 360.49          | 31.67    | 1.00%                 | 31.68    | 1.35%                 | 1.09%     |
| Garden (CIF)     | 8         | 53.75    | 10905.84        | 53.75    | 2.15%                 | 53.77    | 3.19%                 | 0.24%     |
|                  | 16        | 47.52    | 6315.04         | 47.52    | 2.98%                 | 47.53    | 4.45%                 | 0.39%     |
|                  | 24        | 39.66    | 2943.66         | 39.66    | 3.94%                 | 39.67    | 4.84%                 | 0.71%     |
|                  | 32        | 31.91    | 995.98          | 31.91    | 3.83%                 | 31.91    | 4.18%                 | 1.24%     |
| Parkrun (1280×720)| 8       | 53.83     | 149152.3        | 53.83    | 0.76%                 | 53.83    | 1.75%                 | 0.20%     |
|                  | 16        | 46.58    | 90489.58        | 46.58    | 1.22%                 | 46.59    | 3.15%                 | 0.34%     |
|                  | 24        | 38.10    | 39474.4         | 38.10    | 1.56%                 | 38.11    | 2.69%                 | 0.77%     |
|                  | 32        | 30.83    | 9357.64         | 30.83    | 1.20%                 | 30.84    | 1.88%                 | 1.13%     |

Table 9  The percentages of improvement using four scan methods.

| QP  | FrameZero | MBZero | ClusterZero (NC=2) | ClusterZero (NC=5) |
|-----|-----------|--------|--------------------|--------------------|
| 8   | 14.57%    | 14.71% | 0.01%              | 70.71%             |
| 16  | 17.69%    | 17.00% | 0.04%              | 65.27%             |
| 24  | 23.07%    | 20.44% | 0.62%              | 55.87%             |
| 32  | 23.99%    | 21.65% | 0.69%              | 44.66%             |

Table 10  The average increase of computational complexity of AZDS.

| QP  | Computational complexity |
|-----|--------------------------|
| 8   | 6.45%                    |
| 16  | 5.60%                    |
| 24  | 3.76%                    |
| 32  | 2.43%                    |

5. Conclusion

An adaptive zero-coefficient distribution scan (AZDS) is proposed for inter mode coding for H.264/AVC baseline profile. It is found that Run Before affect the coding efficiency most among all five syntax elements of CA VLC. The zero-coefficient distribution of 4x4 blocks of the previous frame is employed to form the new scan methods. The necessary side information bits are also discussed. The corresponding new VLC tables are also provided. In the future, adaptive SSTC for selecting available scan methods will be
incorporated for slow motion sequences and the VLC tables of side information will be performed for low bit-rate cases. Furthermore, AZDS combined with the RDO algorithm can also be applied. We will also focus on improving the coding efficiency of context-adaptive binary arithmetic coding (CABAC) coder by applying the above methodology.

Acknowledgments

This work was sponsored by the National Science Council of Taiwan, under project NO NSC-97-2221-E-006-131-MY3.

References

[1] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC),” Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.
[2] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol.13, no.7, pp.560–576, July 2003.
[3] “Information technology—Generic coding of audio—visual objects Part2: Visual,” ISO/IEC, ISO/IEC 14496-2 (MPEG-4 Video), April 1999.
[4] X. Fan, Y. Lu, and W. Gao, “A novel coefficient scanning scheme for directional spatial prediction-based image compression,” IEEE ICME, pp.II/557–560, July 2003.
[5] Y.-L. Lee, K.-H. Han, D.-G. Sim, and J. Seo, “Adaptive scanning for H.264/AVC intra coding,” ETRI J., vol.28, pp.668–671, Oct. 2006.
[6] D.-Y. Kim, D.-K. Kim, and Y.-L. Lee, “Adaptive scanning using pixel similarity for H.264/AVC,” IEICE Trans. Fundamentals, vol.E90-A, no.5, pp.1112–1114, May 2007.
[7] B.-D. Choi, J.-H. Kim, and S.-J. Ko, “Adaptive coefficient scanning based on the intra prediction mode,” ETRI J., vol.29, pp.694–696, Oct. 2007.
[8] Y. Tao, Y. Peng, and Z. Liu, “More scanning patterns for entropy coding for H.264,” ISPACS 2007, pp.490–493, Nov. 2007.
[9] Y.J. Yoo, S. Jeong, and J.K. Han, “Adaptive scan pattern for quantized coefficients in intra coding of H.264,” IEICE Trans. Inf. & Syst., vol.E92-D, no.4, pp.750–752, April 2009.
[10] Q.-C. Sun, X.-Y. Wu, and L. Yu, “An adaptive coefficient scanning scheme for inter-prediction coding,” Picture Coding Symposium 2007, Lisbon, Portugal, 2007.
[11] L. Yu, F. Yi, J. Dong, and C.X. Zhang, “Overview of AVS-video: Tools, performance and complexity,” Visual Communications and Image Processing (VCIP), pp.679–690, Beijing, July 2005.
[12] Joint Video Team (JVT) Reference Software [Online]. Available: http://bs.hhi.de/suehringtml/download/

Jing-Xin Wang received his M.S. degree from Chung Hua University in 2003. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering at National Cheng Kung University. His research interests include video and image compression, digital audio and video signal processing and MPEG-4/H.264/SVC video codec.
Alvin W.Y. Su was born in Taiwan in 1964. He received the B.S. degree in control engineering from National Chiao-Tung University (NCTU), Taiwan, in 1986. He received the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University, Brooklyn, NY, in 1990 and 1993, respectively. From 1993 to 1994, he was with Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford, CA. From 1994 to 1995, he was with Computer Communication Lab of the Industrial Technology Research Institute (CCL, ITRI.), Taiwan. In 1995, he joined the Department of Information Engineering and Computer Engineering at Chung-Hwa University, where he serves as an Associate Professor. In 2001, he joined the Department of Computer Science and Information Engineering, National Cheng-Kung University (NCKU). His research interests include digital audio signal processing, physical modeling of acoustic musical instruments, audio codec, human computer interface design, video and color image signal processing, MPEG/H.264 video codec, VLSI signal processing, ESL tool design and multi-core CPU. Dr. Su is a Member of IEEE Computer Society and Signal Processing Society. He is also a Member of Audio Engineering Society.