# LOSSLESS VIDEO CODING USING VARIABLE BLOCK-SIZE MC AND 3D PREDICTION OPTIMIZED FOR EACH FRAME

*Ichiro Matsuda, Taichiro Shiodera and Susumu Itoh*

Department of Electrical Engineering, Faculty of Science and Technology,
Science University of Tokyo
2641 Yamazaki, Noda-shi, Chiba 278-8510, JAPAN
Tel: +81 4 7124 1501 (ex.3722); Fax: +81 4 7124 9367
E-mail: matsuda@itohws01.ee.noda.sut.ac.jp

## ABSTRACT

This paper proposes an efficient lossless coding scheme for video signals. The coding scheme utilizes a novel block-adaptive 3D prediction method which predicts a video signal at each pel based on both the current frame and the motion-compensated previous frame. The resulting prediction errors are encoded using a kind of context-adaptive arithmetic coding. In order to improve coding efficiency, not only prediction coefficients of the 3D linear prediction but also motion vectors in all the blocks are iteratively optimized for each frame so that a coding rate of prediction errors can have a minimum. Moreover, a variable block-size motion compensation technique is employed for efficient representation of motion information. Experimental results show that coding rates of the proposed scheme are 18–55% lower than those of the JPEG-LS based intra-frame coding scheme.

## 1. INTRODUCTION

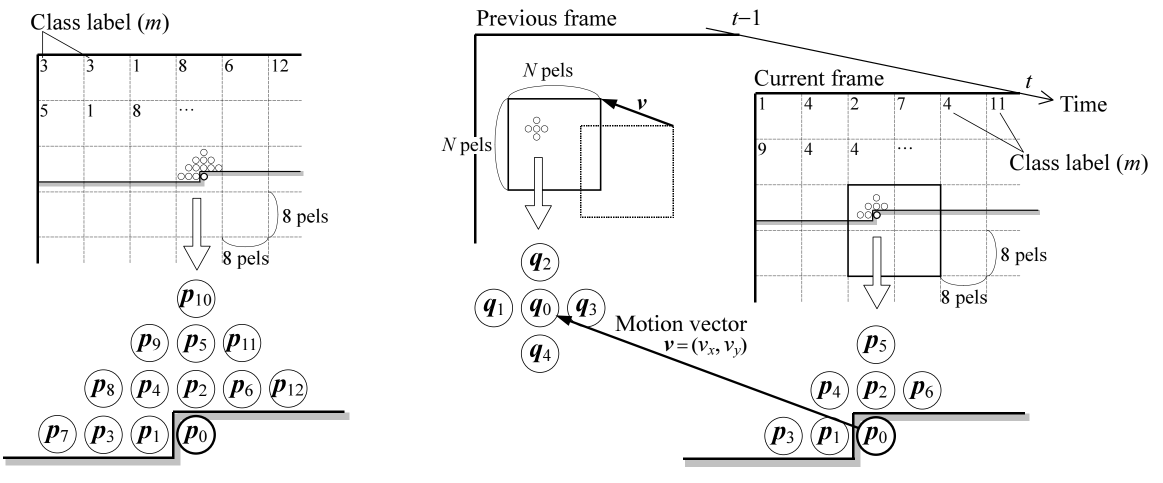
Lossless video coding is useful for specific applications, such as digital archiving and studio production, and several coding schemes have been proposed in recent years. Algorithms used in these schemes can be almost categorized into two types: intra-frame coding and inter-frame coding. Though intra-frame coding has an advantage in easy random access to each frame, its coding performance is usually worse than that of inter-frame coding because it cannot use temporal correlation between adjacent frames. In addition, if decoding speed becomes fast enough, lack of the random access ability in inter-frame coding can be overcome by periodical insertion of key-frames, or intra-coded frames. Therefore, inter-frame lossless coding which has great potential for efficiency improvement is attracting more attention than ever before. For example, an extension of the JPEG-LS standard [1] to lossless video coding based on context-adaptive intra/inter-frame prediction has been proposed in [2]. Moreover, motion compensation (MC) has been employed in [3] to improve efficiency of inter-frame prediction. A recent paper [4] has described a novel 3D prediction technique based on MC. In this technique, a 3D linear predictor, which predicts a video signal using both the current frame and the motion-compensated previous frame, is optimized at each pel in a minimum mean square error (MMSE) criterion. However, such a pel-by-pel optimization approach, which is often called backward adaptation, increases complexity of not only encoding but also decoding operations because the identical optimization process must be carried out at both encoder and decoder sides. Furthermore, the MMSE-based design of a predictor is not necessarily optimum in lossless compression where reduction of a coding rate is the primary purpose [5].

From these points of view, this paper proposes an efficient lossless video coding scheme based on variable block-size MC and block-adaptive 3D prediction. The basic idea of the scheme is derived from our adaptive prediction method which has been developed for still image coding [6]. In the method, a set of linear predictors are designed optimally for a given image so that a coding rate of prediction errors can have a minimum, and an appropriate predictor is selected block-by-block from the set. Such an optimization approach, which uses global information unavailable at the decoder and is called forward adaptation, generally attains more stable and better coding performance than backward adaptation in spite of need of side information on the adaptation. Besides, it allows a fast decoding operation because the predictor optimization process is no longer necessary at the decoder side. In the proposed scheme, not only prediction coefficients but also motion vectors are iteratively optimized for each frame to realize better collaboration between MC and the 3D prediction. Moreover, a quadtree based variable block-size MC technique is introduced to transmit the motion vectors efficiently.

## 2. MC BASED 3D PREDICTION

The proposed scheme encodes all the frames in their temporal order. The current frame to be encoded is partitioned into square blocks composed of $8 \times 8$ pels and each block is classified into one of twelve classes ($m = 1, 2, \cdots, 12$). Each class has an individual predictor which is optimized for blocks belonging to the same class. For the first frame, 12-th order 2D intra-frame prediction is employed as shown in Figure 1 (a). On the other hand, MC based 3D inter-frame prediction is applied to the other frames as shown in Figure 1 (b). In both figures, $\boldsymbol{p}_0 = (x, y)$ is the current pel to be predicted and the remaining $\boldsymbol{p}_k$s ($k = 1, 2, \cdots$) are pels used for the prediction. In the case of the 3D prediction, not only the six pels $\boldsymbol{p}_k$s ($k = 1, 2, \cdots, 6$) in the current frame but also five pels $\boldsymbol{q}_k$s ($k = 0, 1, \cdots, 4$) in the previous frame are used for the prediction. In addition, positions of $\boldsymbol{q}_k$s are shifted according to a motion vector $\boldsymbol{v} = (v_x, v_y)$ which is detected in an MC-block composed of $N \times N$ pels. This means for example $\boldsymbol{q}_0 = \boldsymbol{p}_0 + \boldsymbol{v}$. Therefore a prediction error $e$ produced by the 3D predictor of the $m$-th class is expressed as:

$$e = S_t(\boldsymbol{p}_0) - \sum_{k=1}^{6} a_m(k) \cdot S_t(\boldsymbol{p}_k) - \sum_{k=0}^{4} a_m(k+7) \cdot S_{t-1}(\boldsymbol{q}_k), \quad (1)$$




(a) Intra-frame prediction (b) Inter-frame prediction

**Figure 1** Disposition of pels for the prediction.

where $S_t(\boldsymbol{p}_k)$ and $S_{t-1}(\boldsymbol{q}_k)$ are already encoded video signals at the pels $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ respectively, and $a_m(k)$s ($k = 1, 2, \cdots, 11$) are prediction coefficients of the $m$-th predictor. It should be noted that the above block-adaptive 3D prediction includes 6-th order 2D intra-frame prediction as a special case where all of the prediction coefficients $a_m(k)$s ($k \geq 7$) for the pels $\boldsymbol{q}_k$s are set to be zero. Therefore, adaptive switching of intra/inter-frame prediction as proposed in [2] is not necessary in the proposed scheme.

## 3. CODING OF PREDICTION ERRORS

After conducting the above prediction, the prediction error $e$ at each pel is mapped to a non-negative integer $E$ which we call an error index hereafter. Values of the error index $E$ are allocated to all the possible values of $e$ in increasing order of $|e|$. Then a simple context modeling method [6] is performed to estimate probability distribution of the error index. This method is based on non-linear quantization of the following parameter $U$ which is calculated for the current pel $\boldsymbol{p}_0$:

$$U = \sum_{k=1}^{6} E_t(\boldsymbol{p}_k) + \sum_{k=0}^{4} E_{t-1}(\boldsymbol{q}_k), \qquad (2)$$

where $E_t(\boldsymbol{p}_k)$ and $E_{t-1}(\boldsymbol{q}_k)$ are the error indices already obtained at the pels $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ respectively. In the case of the 2D intra-frame prediction, the second term in Eq.(2) is not added at all. Each quantization level of $U$ corresponds to one of sixteen contexts as shown in Figure 2. In this example the $n$-th context is allocated to the current pel $\boldsymbol{p}_0$ and the estimated probability distribution of $E = E_t(\boldsymbol{p}_0)$ is given by the following generalized Gaussian function:

$$P_n(E) = \alpha_n \cdot \exp\left( - \left| \sqrt{\frac{\Gamma(3/c_n)}{\Gamma(1/c_n)}} \cdot \frac{E}{2\sigma_n} \right|^{c_n} \right), \qquad (3)$$

where $\Gamma(\cdot)$ is the gamma function and $\alpha_n$ is a normalizing factor that makes the sum total of the probability equal to one. $\sigma_n$ and $c_n$ represent parameters which control properties of the generalized Gaussian function $P_n(E)$ for the $n$-th context. The parameter $\sigma_n$, which corresponds to a standard deviation of prediction errors in the $n$-th context, is fixed for all video sequences in this paper. On

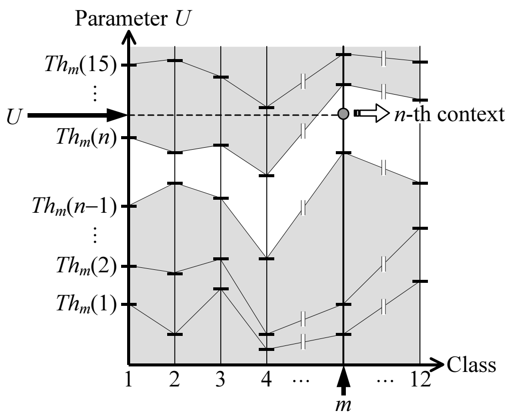



**Figure 2** Context modeling for adaptive arithmetic coding of prediction errors.

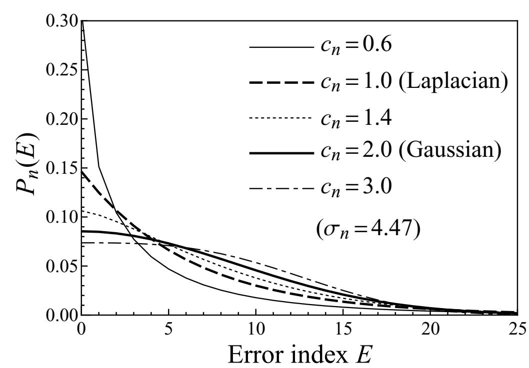



**Figure 3** Relationship between the shape parameter $c_n$ and $P_n(E)$ in the case of $n = 10$.

the other hand, a value of $c_n$, which is called a shape parameter, is optimized to fit $P_n(E)$ to the actual probability distribution in the $n$-th context. Figure 3 shows examples of $P_n(E)$ for different values of $c_n$. Moreover, thresholds $\{Th_m(1), Th_m(2), \cdots, Th_m(15)\}$ used in the above sixteen-level quantization are optimized in each class. The details of these optimization procedures will be described in the next section. Finally, a value of the error index $E = E_t(\boldsymbol{p}_0)$ is entropy coded using the range coder [7] with $P_n(E)$, which is a multi-symbol arithmetic coder.

## 4. OPTIMIZATION OF CODING PARAMETERS

In the proposed lossless video coding scheme, parameters listed below are optimized for each frame and transmitted to the decoder as side information.

- Motion vector $\boldsymbol{v}$ for each MC-block composed of $N \times N$ pels in the case of the 3D inter-frame prediction.
- Class label $m$ for each block composed of $8 \times 8$ pels.
- Prediction coefficients $a_m(k)$s for each class.
- Thresholds $\{Th_m(1), Th_m(2), \cdots, Th_m(15)\}$ for each class.
- Shape parameter $c_n$ for each context.

Optimization of these coding parameters is carried out by iteratively minimizing the following cost function:

$$J = -\sum_{\boldsymbol{p}_0} \log_2 P_n(E = E_t(\boldsymbol{p}_0)). \tag{4}$$

This cost function represents considerably accurate approximation of the total amount of bits which would be given by the above-mentioned context-adaptive arithmetic coding of prediction errors. Concrete procedures for the optimization are as follows.

(1) Detect an initial motion vector $\boldsymbol{v}$ in each MC-block composed of $N \times N$ pels by using MMSE-based block matching.
(2) Classify each block composed of $8 \times 8$ pels into one of twelve classes and design an initial predictor for each class.
(3) Choose two prediction coefficients $a_m(i)$ and $a_m(j)$ randomly, and carry out the partial optimization by varying their values gradually. Repeat this operation a certain number of times in each class.
(4) Optimize the thresholds $\{Th_m(1), Th_m(2), \cdots, Th_m(15)\}$ in each class by using the dynamic programming technique.
(5) Re-select the optimum value of the shape parameter $c_n$ from among sixteen kinds of prearranged values in each context.
(6) Re-classify all the blocks by selecting the optimum predictor, or the optimum class in each block.
(7) Refine a motion vector $\boldsymbol{v}$ of each MC-block by selecting the best position from among its four neighboring search points.
(8) Repeat the above procedures (3)–(7) until all the coding parameters converge in each frame.

## 5. VARIABLE BLOCK-SIZE MC

In the foregoing sections, the size of MC-blocks is expressed as $N \times N$ pels. Obviously, adoption of the smaller block-size may improve efficiency of the 3D inter-frame prediction, however it increases amount of side information on the motion vectors. In order to investigate the relationship between the block-size and the total coding efficiency, we tested the proposed coding scheme for different values of $N$. Motion vector components $(v_x, v_y)$ are restricted to integer-pel accuracy within the search range of $[-15, +15]$. $v_x$ and $v_y$ are differentially encoded using the H.263-like median prediction [8] and the range coder. Probability distribution of a differential motion vector $(\Delta v_x, \Delta v_y)$ required for the range coder is modeled by a 2D generalized Gaussian function $P_v(\Delta v_x, \Delta v_y)$ as shown in Figure 4. Table 1 indicates the overall coding rates obtained for several monochrome video sequences, each of which is 15 frames long and has a frame-size of $352 \times 288$ pels (CIF). The best result for each sequence is shown in **boldface** in the table. Though coding rates of prediction errors, which do not include any side information and are also shown with parentheses in the table, can be considerably reduced in some sequences by using the small block-size ($N = 8$), the overall coding rates including all of the side information do not decrease due to increase of motion vectors. This suggests that further improvement of coding performance will be possible if the motion information can be represented more efficiently.

Thereupon, we introduce quadtree based variable block-size partitioning into the MC operation. In order to optimize the block-size for MC as well as the other coding parameters, a decision process of the block-size is also integrated into the iterative procedures described in Section 4. $N = 16$ is used as an initial value of the block-size and the coding parameters such as motion vectors and prediction coefficients

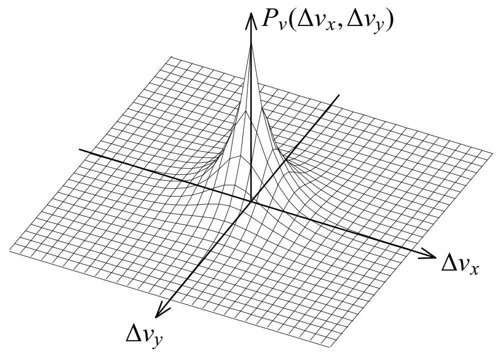



**Figure 4** Example of the 2D generalized Gaussian function $P_v(\Delta v_x, \Delta v_y)$ used for arithmetic coding of motion vectors.

**Table 1** Coding rates for different values of the block-size $N$ of MC-blocks (bits/pel).

| Sequence | $N = 8$ | $N = 16$ | $N = 32$ |
|---|---|---|---|
| Carphone | 2.922 (2.759) | **2.890** (2.806) | 2.897 (2.829) |
| Container | 2.569 (2.481) | 2.552 (2.486) | **2.550** (2.487) |
| Foreman | 3.006 (2.862) | **2.974** (2.894) | 2.983 (2.921) |
| Mobile | 4.118 (3.992) | **4.063** (3.994) | 4.065 (4.002) |
| News | 1.566 (1.475) | **1.548** (1.488) | 1.558 (1.502) |
| Tempete | 3.729 (3.621) | **3.702** (3.634) | 3.703 (3.642) |
| Average | 2.985 (2.865) | **2.955** (2.884) | 2.959 (2.897) |

are determined once through the first iteration. Additionally, motion vectors suitable for the other block-size ($N = 8$ and $N = 32$) are detected by full search block matching with the cost function $J$. Then three-level partitioning based on a quadtree structure is performed in top-down order, that is the order of $N = 32, 16, 8$, and the best combination of the block-size for MC which minimizes the sum of the cost function $J$ and amount of motion information is determined in each area of $32 \times 32$ pels. This decision process of the block-size is carried out every after the procedure (7) described in Section 4, and the other coding parameters are optimized iteratively based on the preceding variable block-size partitioning.

## 6. PERFORMANCE EVALUATION

Table 2 compares coding rates of some lossless coding schemes. 'VBS' means the proposed scheme with the above-mentioned variable block-size MC. 'MMSE' means a coding scheme which is identical to 'VBS' except that the MMSE criterion is used for the optimization procedures instead of the cost function $J$. 'INTRA' is another variant of the proposed scheme, where intra-frame coding is applied to all of the frames. And 'JPEG-LS' is also an intra-frame coding scheme which utilizes the JPEG-LS algorithm [1] on a frame-by-frame basis. In addition meaning of parentheses in this table is the same as in Table 1.

We can see that 'VBS' always outperforms 'MMSE' in terms of coding efficiency. The fact demonstrates that the conventional MMSE-based approach is not optimum in lossless coding as we have pointed out for still images before [5]. Moreover, coding rates of 'VBS' are about 0.01 bits/pel lower than those of the fixed block-size scheme ($N = 16$) shown in Table 1. Figure 5 indicates a result of variable block-size partitioning. It is observed that appropriate block-size is selected for MC according to local motion activity. Again, an average coding rate of 'VBS' is 0.85 and 1.25 bits/pel lower than that of 'INTRA' and 'JPEG-LS' respectively. These large gains prove that the MC-based adaptive 3D prediction proposed in this paper considerably contributes to improvement of coding performance.

**Table 2** Comparison of coding rates (bits/pel).

| Sequence | VBS | MMSE | INTRA | JPEG-LS |
|---|---|---|---|---|
| Carphone | **2.880** (2.789) | 2.945 (2.852) | 3.143 (3.083) | 3.508 |
| Container | **2.551** (2.486) | 2.569 (2.506) | 3.657 (3.596) | 4.140 |
| Foreman | **2.949** (2.862) | 3.000 (2.911) | 3.415 (3.352) | 3.821 |
| Mobile | **4.045** (3.974) | 4.137 (4.063) | 4.933 (4.861) | 5.403 |
| News | **1.544** (1.484) | 1.591 (1.530) | 3.138 (2.925) | 3.420 |
| Tempete | **3.685** (3.620) | 3.752 (3.681) | 4.493 (4.423) | 4.836 |
| Average | **2.942** (2.869) | 2.999 (2.924) | 3.797 (3.706) | 4.188 |

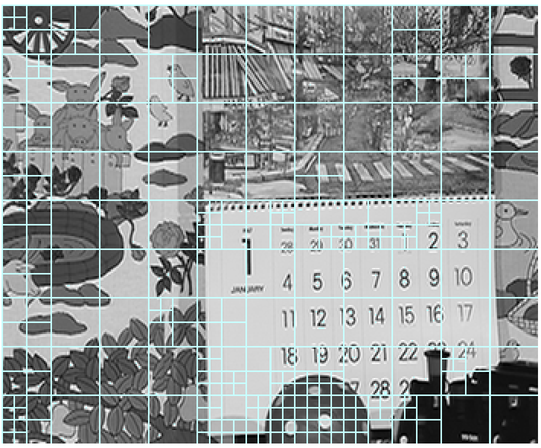


**Figure 5** Result of variable block-size partitioning in the 2nd frame of the Mobile sequence.

## 7. CONCLUSIONS

We have proposed a novel lossless video coding scheme based on variable block-size MC and block-adaptive 3D prediction. In order to improve coding efficiency, several coding parameters are iteratively optimized for each frame so that a coding rate of prediction errors can have a minimum. Simulation results show that coding rates of the proposed scheme are 18–55% lower than those of the JPEG-LS based intra-frame coding scheme.

Since the proposed coding scheme fundamentally stands on forward adaptation and the above iterative optimization process is not necessary at the decoder side, decoding speed is fast enough for practical video applications. For example, our prototype software codec which is implemented by using C language can decode CIF size monochrome sequences at more than 50 frames/second on a personal computer with 3 GHz Intel Pentium 4 processor.

In this paper we have investigated appropriate block-size for MC. With respect to the number of the contexts, it has been already confirmed that sixteen is sufficient for many video sequences. On the other hand, other parameters such as prediction order, the number of predictors and block-size for the adaptive prediction will be studied more closely.

## REFERENCES


[1] ISO/IEC 14495-1, "Information Technology – Lossless and Near-Lossless Compression of Continuous-Tone Still Images: Baseline," Dec. 1999.

[2] K. H. Yang and A. F. Faryar, "A Context-based Predictive Coder for Lossless and Near-Lossless Compression of Video," *Proc. of ICIP-2000*, Vol. I, pp. 144–147, Sep. 2000.

[3] B. Martins and S. Forchhammer, "Lossless Compression of Video Using Motion Compensation," *Proc. of DCC '98*, p. 560, 1998.

[4] D. Brunello, G. Calvagno, G. A. Mian and R. Rinaldo, "Lossless Compression of Video Using Temporal Information," *IEEE Trans. on Image Processing*, Vol. 12, No. 2, pp. 132–139, Feb. 2003.

[5] I. Matsuda, H. Mori and S. Itoh, "Lossless Coding of Still Images Using Minimum-Rate Predictors," *Proc. of ICIP-2000*, Vol. I, pp. 132–135, Sep. 2000.

[6] I. Matsuda, N. Shirai and S. Itoh, "Lossless Coding Using Predictors and Arithmetic Code Optimized for Each Image," *Proc. of VLBV03, Lecture Notes in Computer Science*, Vol. 2849, pp. 199–207, Springer-Verlag, Sep. 2003.

[7] M. Schindler, "A Fast Renormalization for Arithmetic Coding," *Proc. of DCC '98*, p. 572, 1998.

[8] ITU-T SG15, "Video Coding for Low Bitrate Communication," Draft ITU-T Rec. H.263, May 1996.