3D MOTION VECTOR CODING WITH BLOCK BASE ADAPTIVE INTERPOLATION FILTER ON H.264

Hideaki Kimata, Masaki Kitahara, and Yoshiyuki Yashima

NTT Cyber Space Laboratories, NTT Corporation

ABSTRACT

Fractional pel motion compensation generally improves coding efficiency due to more precise motion accuracy and low path filtering effect in generating image at fractional pel positions. In H.264, quarter pel motion compensation is applied, where image at half pel position is generated by 6 tap Wiener filter. And the adaptive interpolation filter technique, which adaptively changes filter characteristics for half pel positions have been proposed. That technique also changes image at quarter pel positions, so it can be exploited to extent motion accuracy to be more precise. In this paper, 3D Motion Vector Coding (3DMVC) technique with Block base Adaptive Interpolation Filter (BAIF) is proposed. This paper also demonstrates the proposed method achieves filter data is successfully integrated into motion vector coding and outperforms the normal H.264.

1. INTRODUCTION

In quarter pel motion compensation of H.264, image at half pel position is generated by 6 tap Wiener filter. And image at quarter pel position is generated by linear interpolation of image at integer or half pel positions. The adaptive interpolation filter technique, which changes Wiener filter characteristics for half pel positions per a frame[1] of a macroblock[2] have been proposed. In this technique, image at half pel and quarter pel positions are adaptively changed. Especially in [1], the coefficients of filter and the filter length are determined so that prediction error of a whole frame may be minimized. This method basically has its sights on reducing aliasing artifacts and quantization artifacts of residual error in the decoded pictures. And in [2], the coefficients of filter are determined per a macroblock so that prediction error of a whole macroblock may be minimized. In addition, this method also introduces asymmetric coefficients set of filter. This makes it possible to represent image at quarter pel positions as image at eighth or lower fractional pel positions. Therefore more precise motion can be represented.

In these methods, there are two problems. One is that the prediction error cannot be minimized per a motion block since a maroblock is divided into from one to sixteen blocks in H.264. And the other is that the overhead of indicating or expressing filter coefficients is not taken into account into the design of entropy coding.

This paper proposes 3D Motion Vector Coding (3DMVC) technique in which the filter indication is integrated into 2D motion vector coding. And it proposes Block base Adaptive Interpolation Filter (BAIF), which achieves more precise motion accuracy per a motion block.

In Section 2, quarter pel MC on H.264 is briefly explained. And in section 3, adaptive interpolation scheme is proposed and in section 4, 3D motion vector coding scheme is proposed. In section 5, the effectiveness of proposed method is demonstrated. This paper closes the summary.

2. QUARTER PEL MC ON H.264

In quarter pel MC, at first, image at horizontally half pel position is generated from image at integer pel positions by 6 tap filter. The default coefficients set of filter is described in Figure 1. And secondly, image at vertically half pel position is generated from at integer or half pel position by the same filter. These two steps generate image at any half pel position. Image at quarter pel positions is generated by linear interpolation from surrounding image at integer or half pel positions.

3. ADAPTIVE INTERPOLATION FILTER

In the proposed Block base Adaptive Interpolation Filter (BAIF) method, the coefficients set of the Wiener filter is changed per a motion block. In H.264, the motion segmentation pattern is defined as illustrated in Figure 2. In 8x8 segmentation pattern, 8x8 block is additionally segmented. Each block has one motion vector, and filter indication. Examples of coefficients set of the filter are illustrated in Figure 3 [2]. FIL1 is the same as the default filter. FIL2 and FIL3 are asymmetric filters, which

$$\{1, -5, 20, 20, -5, 1\}/32$$

Fig.1. Default coefficients set of 6 tap filter

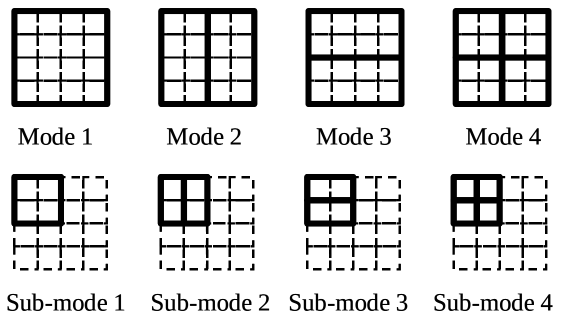

Fig.2. Partitioning modes of a macroblock

FIL1: {8, -40, 160, 160, -40, 8}/256
FIL2: {8, -40, 216, 96, -32, 8}/256
FIL3: {8, -32, 96, 216, -40, 8}/256

Fig.3. Coefficients sets of asymmetric filters

represent quarter pel positions as eighth pel positions. Therefore motion representation whose accuracy is equivalent to eighth pel motion compensation is possible. In general, more precise motion can be represented in this scheme.

4. 3D MOTION VECTOR CODING

Since applying asymmetric filters is equivalent to make motion accuracy more precise, filter indication should be coded with motion vector. When one moving object exists on multiple macroblocks, motion blocks, whose motion information are the same, exist on those macroblocks. In such blocks, the same filters should be applied. Therefore we focus on reducing the overhead of filter indication when difference between adjacent motion blocks is nothing.

4.1. 2D Motion Vector Coding for Non Adaptive Interpolation Filter

So that integrating filter indication to motion vector coding may be easy, motion vector coding itself is changed to 2DVLC. In H.264, each motion parameter, which is horizontal or vertical, is coded by Golomb codes. Each motion parameter is predicted from median of motion vectors of surrounding blocks illustrated in Figure 4. To express the situation that motion information is zero, i.e. both motion parameters are zero, a single code to this state is assigned in 2DVLC. Extending this scheme, variable length codes are assigned to other combination of motion parameters. Figure 5 illustrates the destination point of motion vector from the origin point. And Table 1 shows the codes for combination of motion parameters. Generally they are distinguished to four categories. The first one is the situation of both zero, which is illustrated

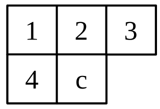

Fig.4. Illustration of adjacent blocks (1,2,3,4), where "c" is the current block

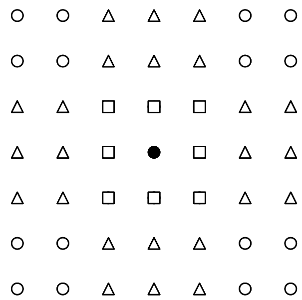

Fig.5. Destination positions of motion vector in quarter pel, where the block circle is the origin point, thus it indicates (0,0) vector

Table 1. Codewords for 2DMVC

MV	Code
(0,0)	1
(-1/4,-1/4), (-1/4,0), (-1/4,1/4), (0,-1/4), (0,1/4), (1/4,-1/4), (1/4,0), (1/4,1/4)	(01000 – 01111)
Counterpart is one of {-1/4,0,1/4}	ExpGolomb code for both horizontal and vertical +1bit flag
Other	ExpGolomb code for both horizontal and vertical

by the black circle in Figure 5. And the second one is both motion parameters are one of {-1/4, 0, 1/4}, which is illustrated by the squares in Figure 5. The third one is either motion parameter is one of {-1/4, 0, 1/4}, which is illustrated by the triangles in Figure 5. The fourth one is for any other combination, which is illustrated by the white circle in Figure 5. In the third category, the bit pattern is the same as that one bit flag to indicate whether the motion parameter {-1/4, 0, 1/4} is horizontal or vertical, is coded after two variable length codes as traditional motion vector coding.

4.2. 3D Motion Vector Coding with Adaptive Interpolation Filter

The coefficients set of filter is determined in motion estimation process, and motion vector is determined at the same time. Since filter characteristics is similar to those of adjacent blocks when motion vector is zero, the

Table 2. Codewords for 3DMVC

MV	Filter	Code
(0,0)	Predicted	1
(-1/4,-1/4), (-1/4,0), (-1/4,1/4), (0,-1/4), (0,1/4), (1/4,-1/4), (1/4,0), (1/4,1/4)	(1,1)	(01000 – 01111)
(-1/4,-1/4), (-1/4,0), (-1/4,1/4), (0,-1/4), (0,1/4), (1/4,-1/4), (1/4,0), (1/4,1/4)	(1,2),(1,3), (2,1),(2,2), (2,3),(3,1), (3,2),(3,3),	(001000 – 001111) + (000 – 111)
Counterpart is one of {-1/4,0,1/4}	Split	ExpGolomb code for both horizontal and vertical +1bit flag
Other	Split	ExpGolomb code for both horizontal and vertical

Table 3. Codewords for Split Filter indication of 3DMVC

Filter	Code
FIL1	1
Other combination	01000 - 01111

Table 4. Simulation conditions

Images	Stefan, Mobile (CIF, 300f)
Framerate	30fps
Quantization	Fixed (Q=4,8,12,16,20,24)
Software	JM1.9+MotionCopy

coefficients set of such filter is forced to be the predicted one from those of adjacent blocks, i.e. the coefficient set of filter is forced to the most existing set among adjacent blocks' illustrated in Figure 4.

In case that the coefficients set of filter is one of {FIL1, FIL2, FIL3} as illustrated in Figure 3, Table 2 shows the codes for combination of motion parameters and filer number. Generally they are distinguished to five categories. The first one is the situation of both zero. And the second one is both motion parameters are one of {-1/4, 0, 1/4}, and both horizontal and vertical filters are FIL1. The third one is both motion parameters are one of {-1/4, 0, 1/4}, but either horizontal or vertical filter is either FIL2 or FIL3. The fourth one is either motion parameter is one of {-1/4, 0, 1/4}. The fifth one is for any other combination. In the fourth category, the bit pattern is the same as that both one bit flag to indicate whether the motion parameter {-1/4, 0, 1/4} is horizontal or vertical, and variable length code of filter indication shown in

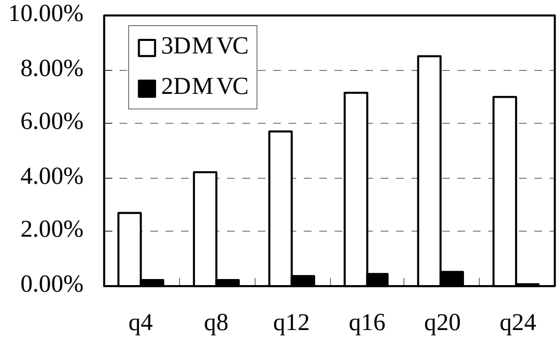

Fig.6. Average bit saving for Mobile, where number of reference pictures is one

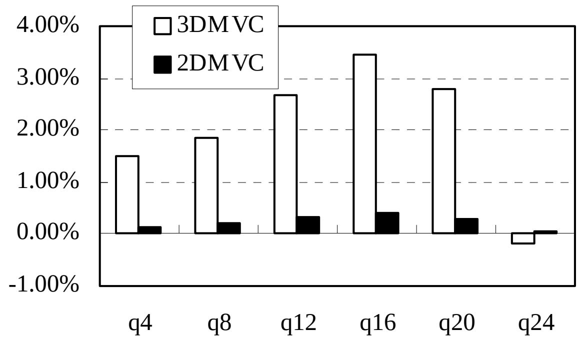

Fig.7. Average bit saving for Stefan, where number of reference pictures is one

Table 3, are coded after two variable length codes for traditional motion vector coding. In the fifth category, the bit patter is the same as that variable length code of filter indication shown in Table 3 is coded after two variable length codes.

Especially in H.264, since the motion vector of the Skip macroblock in Interframe is predicted, the coefficients set of filter is also predicted. And the coefficients sets illustrated in Figure 3 represent eighth pel motion compensation for luma, the motion vector for chroma is calculated in sixteenth pel accuracy corresponding to the determined coefficients set of filter.

5. EXPERIMENTAL RESULTS

The effectiveness of the proposed method, 3DMVC, was evaluated by computer simulation. Table 4 shows the simulation conditions. The used coefficients sets of filter are as illustrated in Figure 3. At motion estimation, both the motion vector and the coefficient set of filter are determined so that motion cost J may be the smallest. The motion cost J is calculated by the equation (1), where R is the total bit number for a motion block and $r(i, j)$ is the decoded image and $o(i, j)$ is the original

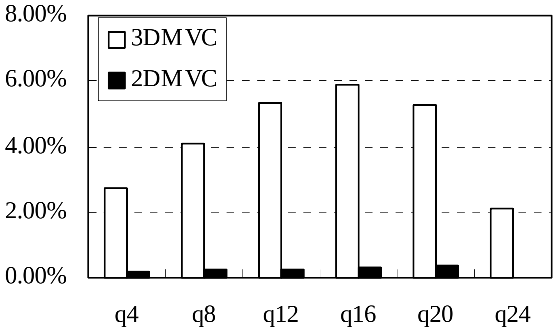

Fig.8. Average bit saving for Mobile, where number of reference pictures is two

image. Every operation was done in block encoding loop, so no optimization was done per a frame or a macroblock.

$$\left.\begin{aligned} J &= SSD + \lambda \times R \\ SSD &= \sum_{i,j} (r(i,j) - o(i,j))^2 \end{aligned}\right\} \quad (1)$$

Figure 6 or Figure 7 illustrates the experimental results of the bit saving rate for Mobile or Stefan sequence when the number of reference picture is one. And Figure 8 illustrates those for Mobile when the number of reference picture is two. PSNR of 2DMVC or 3DMVC is slightly higher than that of the basement at all rates.

For all results, 2DMVC achieves slightly better coding efficiency than the basement (less than 1% range). This is actually due to the effect of bit reduction for motion vector coding when motion vector is zero. And 3DMVC achieves better coding efficiency at almost the all bitrates, especially better at medium bitrates (maximum about 8.5%). Comparing Figure 6 and Figure 8, we can see the coding efficiency become better when the number of reference pictures becomes small. This is thought to be because the variation of motion accuracy for the similar context of image increases when the number of reference pictures increases.

Comparing Figure 6 and Figure 7, we can see the coding efficiency for Mobile is better than that for Stefan. This is thought to be because 3DMVC successfully reduces the bit for filter indication when motion vector is zero. 3DMVC is thought to be effective for stationary motion sequences.

6. SUMMARY

3D motion vector coding (3DMVC) with block base adaptive interpolation filter (BAIF) is proposed. 3DMVC achieves better coding efficiency especially at medium bitrates. It is also demonstrated that 3DMVC is effective for stationary motion sequences.

7. REFERENCES

[1] T. Wedi, “Adaptive Interpolation Filter for Motion Compensated Hybrid Video Coding,” PCS 2001, 2001.
[2] Kei-ichi Chono and Yoshihiro Miyamoto, “Modified Adaptive Interpolation Filter,” JVT-D078, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Meeting, 2002.
[3] G. Y. Yu and C.-T. Chen, “Two-dimension motion vector coding for low bitrate videophone application,” ICIP’95, 1995.