MODIFICATION OF THE MERGE CANDIDATE LIST FOR DEPENDENT VIEWS IN 3D-HEVC

E-G. Mora, J. Jung
Orange Labs
38 Rue du G. Leclerc
92794 Issy Les Moulineaux, France
{elie.mora,joelb.jung}@orange.com

M. Cagnazzo, B. Pesquet-Popescu
Telecom ParisTech
TSI Department
46 rue Barrault, 75634 Paris, France
{cagnazzo,pesquet}@telecom-paristech.fr

ABSTRACT
Standardization activities have recently focused on 3D video with a draft test model for an HEVC-based 3D video coding standard (3D-HEVC). The test model exploits inter-view redundancies by including disparity-compensated prediction (DCP) for efficient dependent view coding. 3D-HEVC also uses the Merge coding mode introduced in HEVC to reduce the cost of motion and disparity parameters. However, the candidates in the Merge list are mostly motion vectors, and although there is a multiview candidate in the list, it is preferred to be a motion vector rather than a disparity vector. Consequently, DCP is often costly, and motion-compensated prediction (MCP) remains largely preferred.

In this paper, we propose a tool that reduces the cost of DCP by modifying the Merge candidate list to always include a disparity vector candidate. Two methods are proposed: the new candidate is either added in the secondary or in the primary list of candidates. Average bitrate reductions of -0.6% for dependent views, and -0.2% for coded and synthesized views are reported. Both methods were presented at the 2nd JCT3V meeting and method 2 was adopted in both the 3D-HEVC working draft and software.

Index Terms— 3D-HEVC, dependent view coding, disparity-compensated prediction, Merge candidate list

1. INTRODUCTION
New 3D multimedia services, such as 3D television [1] (3DTV) or free viewpoint television [2] (FTV) created a need for a 3D video standard that supports multiview video (MVV) and multiview video plus depth (MVD) formats. Indeed, while 3D has not yet met its expected success, autostereoscopic video, enabled by MVV and MVD, offers more comfortable viewing conditions, and is thus considered to be the future of 3D. In 2011, MPEG answered this need by issuing a Call for Proposals on 3D video coding technologies. Later, a joint ISO / ITU-T team called JCT3V, drafted the test model for an AVC-based (3D-AVC) and an HEVC-based (3D-HEVC) 3D coding standard.

3D-HEVC exploits spatial, temporal, inter-component and inter-view redundancies to efficiently encode the 3D video. Inter-view redundancies are in particular exploited by disparity-compensated prediction (DCP) [3], introduced in the MVC standard [4]. DCP enables having, for a currently coded frame, reference frames from different views at the same time instant. The vector of a prediction unit (PU) pointing to a PU in a different view is called a disparity vector (DV).

3D-HEVC includes two ways to determine the motion / disparity parameters (vectors + reference indices) of a PU: Advanced Motion Vector Prediction (AMVP), and the Merge coding mode [5]. Both methods establish lists of vector predictors, and are aimed at reducing the signalling cost of the motion / disparity parameters. While a DV candidate is always present in the AMVP candidate list, the Merge list most of the times contain only motion vectors (MVs). Although there is a multiview candidate in the list, it is always preferred to be a MV rather than a DV. And since the Merge mode is the most efficient coding mode for inter-coded PUs, this asymmetry between the number of DV and MV predictors in the Merge list highly penalizes DCP which remains largely less selected than motion-compensated prediction (MCP).

While numerous tools proposed in HEVC and in 3D-HEVC try to modify the Merge candidate list to achieve coding gains, no tools are designed to populate the list with more DV candidates to reach a better equilibrium between DCP and MCP. In this paper, we propose to modify the Merge candidate list by inserting a new candidate which is always a DV. The DV candidate is inserted either in the secondary (method 1) or the primary (method 2) list of Merge candidates. Both methods were presented at the 2nd JCT3V meeting and method 2 was adopted in both the 3D-HEVC working draft and the software [6].

The rest of this paper is organised as follows: Section 2 presents the state of the art in the Merge coding mode of 3D-HEVC. Section 3 describes our proposed method, and its coding results are given in Section 4. Section 5 concludes this paper, while underlining the possibilities for future work.
2. STATE OF THE ART

The Merge coding mode in 3D-HEVC allows a PU to inherit the motion / disparity parameters from a neighboring PU. Motion / disparity parameters from different neighboring PUs form the Merge candidate list. Only the index of the most coding efficient candidate is sent in the bitstream, along with an optional PU residual. Merge mode thus creates contiguous motion / disparity areas at a minimal cost.

2.1. Merge candidate list in 3D-HEVC

3D-HEVC inherits the Merge candidate list of HEVC [7], and adds, only for dependent views, a multiview candidate [8] in the first position of the list. The HEVC list consists, in order, of four spatial candidates, corresponding to the following positions: left, above, above-right, below-left (and potentially above-left in case one of the earlier position is not available), and one temporal candidate. A pruning process is performed within the spatial candidates to remove redundant vectors [9].

If some of these 6 candidates are unavailable (the PU corresponding to the position falls outside the slice, or is Intra-coded, or the candidate is redundant), a secondary list of candidates is computed. These candidates are then appended to the list so that the total number of candidates is always 6. The candidates in that secondary list are, in order, combined candidates from mixed L0 and L1 vectors of the different primary candidates, and zero-vector candidates, each having a different reference index.

The multiview candidate is computed in the following manner: first, a DV is derived for the current PU. In previous draft versions of 3D-HEVC, a depth map estimate was maintained for each view and the DV was derived from the highest depth value covered by the current PU in the estimate. Currently in 3D-HEVC, to reduce complexity, a simple neighbour search for a DV is performed. The DV allows finding a reference PU in the main view that corresponds to the current PU in the side view. The motion vector of that reference PU is then set as the multiview candidate. If the reference PU is intra-coded or falls outside the slice, the DV itself is set as the multiview candidate. Thus, there is always a temporal preference for this candidate, and consequently, the Merge list is in most cases composed only of MVs.

2.2. Tools that modify the Merge candidate list construction

Several tools that modify the Merge candidate list construction in 3D-HEVC were proposed, to either achieve coding gains, reduce complexity, or lower memory consumption: In [10], the primary candidate list is checked, and the first DV candidate found is used to compute, by adding a positive and a negative offset, two more DV candidates which will then be added to the list. However this requires having a DV in the primary list to begin with in order to work, which is not a frequent case. Consequently, the coding gains are small. In [11], a dynamic re-ordering of the Merge index is performed to reduce its coding cost. An histogram is established and updated to follow the distribution of the merge indices. A conversion table can then be determined from the histogram, and the indices are re-ordered accordingly before CABAC encoding. The coding gains however are not high enough to justify the usage and the frequent updates of the histogram and conversion tables, which increase complexity. The Merge pruning process can also be changed, like in [12] where a comparison between the inter-view candidate and the first two spatial candidates is added. This method was adopted in 3D-HEVC although the gains reported remain relatively small. For depth PU coding, the first Merge candidate was modified in [13] to refer to merging with the co-located texture PU as the texture and depth motion information are highly correlated. A -1.1% bitrate reduction was reported for coded+synthesized views. Hence, this Motion Vector Inheritance tool was adopted in 3D-HEVC.

Tools that affect the Merge candidate list construction were also proposed in HEVC. In [14], the temporal candidate (TMVP) position is changed from the center of the co-located PU to the bottom-right position. Significant bitrate reductions of -0.9% were reported and thus, the method was adopted in HEVC. In [15], it was noticed that the combined candidates could not be constructed for uni-predicted PUs. Hence, two refined candidates computed from the first Merge candidate are constructed and added to the secondary list of candidates to replace the combined. Coding gains were not significant enough however to favor adoption.

All these methods try to improve the candidate list construction but with no particular intention to favor the DCP selection in the process. We propose, as described in the next section, a novel method to favor DCP by inserting a DV candidate in the Merge list.

3. PROPOSED METHOD

3.1. Preliminary study

Our work is based on the observation that DCP is not often selected for coding CUs in the HTM encoder. Also, Merge mode was observed to be often selected for coding PUs. This can be seen in Figure 1(a) which shows parts of a B-frame of the Kendo sequence coded with the reference HTM encoder. The PUs coded using MCP are shown in grey (Merge-SKIP) and green (Inter). PUs coded using DCP are shown in light pink (Merge-SKIP) and dark pink (Inter). Blue PUs are coded in Intra. We can clearly see that Merge mode is selected often, and that DCP coded PUs are not numerous.

Table 1 gives the percentages of Merge coded PUs, DCP coded PUs, and DCP coded PUs in Merge mode, in dependent texture and depth views, averaged across four QPs, of
seven MPEG sequences. The software version used and the test conditions (including QPs) followed in order to get these results are the same as the ones used to evaluate the coding results of our method, as described later in Section 4.1. These results confirm the assertion that Merge mode is selected often. It is actually selected for 92% of PUs. This is due to the fact that Merge mode is very efficient at reducing the cost of motion / disparity parameters as only a simple index is sent in the bitstream. Table 1 also shows that only 16% of PUs use DCP, and they are also most often coded in Merge mode (13%). However, DCP can yield a better hypothesis for a given PU than MCP, in case there is little disparity between views or if there is fast motion in the video. This better prediction reduces the distortion (D) when evaluating a Lagrangian cost (J) for DCP, as per the following equation, where R is the rate and $\lambda$ is the weight of R relative to D:

$$ J = D + \lambda R $$  

(1)

However, not having a DV predictor in the Merge list increases the rate needed to code the PU with DCP since the only option left is AMVP which requires sending a motion vector residual. MCP, while maybe not yielding a lower distortion value, requires a lower rate due to the fact that there are numerous MV predictors in the Merge list and signaling the motion parameters only costs an index. Consequently MCP is chosen more often since its Lagrangian cost is smaller, but if a DV predictor was added in the Merge list, as proposed in this work, the required rate for DCP coding would be decreased, hence increasing the selection of DCP and achieving coding gains.

### 3.2. Method description

When computing the multiview candidate in the Merge candidate list, a DV, which points to a reference block in the base view, is derived. The multiview candidate is set as the MV of that reference block, and if (and only if) that MV does not exist, it is set as the DV. We propose to insert that DV as a new interview candidate in the Merge list along side the multiview candidate if the latter turned out to be a MV.

Two insertion methods are proposed. In method 1, the candidate is inserted in the secondary list along with the combined and the zero vector candidates. If any of the first five candidates (multiview + four spatial) in the primary list is unavailable, the interview candidate is inserted after the final spatial candidate (before the temporal) to complete the list. If more primary candidates are unavailable, the combined and zero vector candidates are then appended to the list, as it is normally done.

In method 2, the candidate is inserted in the primary list, in the 5th position, shifting the final spatial candidate to the 6th position. The temporal candidate is hence pushed out of the primary list and into the secondary list. It is the first candidate in the secondary list to be appended back in the primary list if some candidates are unavailable. Figure 2 illustrates these two methods.

In both methods, before inserting the interview candidate, a redundancy check with all candidates preceding it in the list is performed for better coding efficiency. Note that the insertion positions in both methods have been set empirically as those positions gave out the most coding gains on average.
4. EXPERIMENTAL RESULTS

4.1. Experimental setting

We have implemented our two proposed methods in HTM-4.1 [16]. We have strictly followed the common test conditions (CTCs) defined by JCT3V [17]. A GOP of 8 was considered with an Intra period of 24. Four QP combinations for texture and depth (respectively) were considered: (25;34), (30;39), (35;42) and (40;45) to conform to CTCs. We have tested the two methods on seven sequences defined in the CTCs (1920×1088 and 1024×768). Experiments were done on 10 seconds of video length. Each sequence is composed of 3 texture and 3 depth views (one central base view and two side views). After encoding, 3 intermediate views were synthesized between the left and the center view, and another 3 between the center and the right views. PSNR on synthesized values were computed in respect to synthesized views rendered with uncompressed original texture and depth views. Coding gains are measured with the Bjontegaard delta (BD-Rate) metric [18].

4.2. Coding results

Table 2 and Table 3 give the coding gains (negative values are gains) and runtimes obtained with method 1 and method 2 respectively for each tested sequence. These results are summarized in Table 4 which also gives the average results if the redundancy check preceding the insertion of the interview candidate in the list is removed. In these tables, the “Video” column shows the gains on the central (0) and on the two side views (1 and 2) and averages these results. The “Synth.” column gives results on the 6 synthesized views (the bitrate considered is the sum of the 3 texture and depth bitrates, and the PSNR is the average PSNR of all 6 synthesized views). The “Coded+synth.” result is the same as in the previous column except that the PSNR considered is the average PSNR of the 6 synthesized views and the 3 coded texture views.

Table 2. Coding results per sequence, in %, with method 1

<table>
<thead>
<tr>
<th>Sequence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>Avg</th>
<th>Coded</th>
<th>+Synt</th>
<th>Runtimes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enc</td>
<td>Dec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balloons</td>
<td>0.0</td>
<td>-0.6</td>
<td>-0.3</td>
<td>-0.2</td>
<td>-0.2</td>
<td>96</td>
<td>102</td>
</tr>
<tr>
<td>Kendo</td>
<td>0.0</td>
<td>-0.6</td>
<td>-0.4</td>
<td>-0.2</td>
<td>-0.1</td>
<td>-0.2</td>
<td>100</td>
</tr>
<tr>
<td>Newspaper</td>
<td>0.0</td>
<td>-0.3</td>
<td>-0.3</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>100</td>
</tr>
<tr>
<td>GT Fly</td>
<td>0.0</td>
<td>-1.2</td>
<td>-1.0</td>
<td>-0.3</td>
<td>-0.2</td>
<td>-0.3</td>
<td>97</td>
</tr>
<tr>
<td>Poznan Hall2</td>
<td>0.0</td>
<td>0.2</td>
<td>-0.5</td>
<td>-0.1</td>
<td>-0.1</td>
<td>-0.1</td>
<td>97</td>
</tr>
<tr>
<td>Poznan Street</td>
<td>0.0</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>90</td>
</tr>
<tr>
<td>Dancer</td>
<td>0.0</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>97</td>
</tr>
<tr>
<td>Average</td>
<td>0.0</td>
<td>-0.5</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>97</td>
</tr>
</tbody>
</table>

Table 3. Coding results per sequence, in %, with method 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Video</th>
<th>Synt.</th>
<th>Coded</th>
<th>+Synt</th>
<th>Runtimes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enc</td>
<td>Dec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>0.0</td>
<td>-0.5</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>M1-NO RC</td>
<td>0.0</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.1</td>
<td>-0.1</td>
</tr>
<tr>
<td>M2</td>
<td>0.0</td>
<td>-0.6</td>
<td>-0.6</td>
<td>-0.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>M2-NO RC</td>
<td>0.0</td>
<td>-0.5</td>
<td>-0.4</td>
<td>-0.2</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

Table 4. Coding results when the redundancy check is removed (NO RC) in method 1 (M1) and 2 (M2)

accompanyed by a 3% (respectively 4%) encoder runtime reduction. No gains are achieved on the central view since our method is not applied there. Table 4 shows that a decrease in coding efficiency is achieved if the redundancy check is removed, with no decrease in encoder and decoder runtimes compared to the original version.

4.3. Results interpretation

The gains obtained result from an increase in DCP selection. Inserting a DV into the Merge candidate list reduces the rate needed for DCP coding and favors its selection, especially if there is small disparity between views (interview redundancies are much higher, and DVs can point to a better hypothesis) or if there is fast motion in the video (MVs are not able to correctly predict PUs).

Figure 1(b) indeed shows an increase in DCP coded PUs compared to Figure 1(a). This is confirmed in the numerical results of Table 5, which shows, for both methods, an increase of 1.2% and 1.4% on average in the percentage of DCP-coded PUs and DCP-coded PUs using Merge mode.

One can argue about the complexity of the proposed methods regarding the number of redundancy checks to be performed before insertion. The purpose of this redundancy check is to avoid having a redundant DV candidate in the list which will either push potentially better primary candidates further down the list, while increasing their indices, and hence their signalling cost, in the process, or take the place of other, potentially better, secondary candidates which will not even be evaluated. The maximum number of checks equals
Table 5. Percentage increase of DCP-coded PUs and DCP-coded PUs using Merge mode in the two methods

<table>
<thead>
<tr>
<th>Sequence</th>
<th>DCP increase</th>
<th>DCP-Merge increase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
<td>M2</td>
</tr>
<tr>
<td>Kendo</td>
<td>1.0</td>
<td>1.1</td>
</tr>
<tr>
<td>Newspaper</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>Balloons</td>
<td>1.1</td>
<td>1.0</td>
</tr>
<tr>
<td>Dancer</td>
<td>1.8</td>
<td>1.7</td>
</tr>
<tr>
<td>GT Fly</td>
<td>2.9</td>
<td>2.7</td>
</tr>
<tr>
<td>Poznan Hall2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Poznan Street</td>
<td>0.7</td>
<td>0.9</td>
</tr>
<tr>
<td>Average</td>
<td>1.2</td>
<td>1.2</td>
</tr>
</tbody>
</table>

The two methods also brought small encoder runtime reductions of 3 and 4%. This is because inserting a DV candidate in the Merge list means constructing one less secondary candidate, which is a complex process since it involves mixing in different vectors to construct combined candidates or looping around all reference indices to construct zero-vector candidates. Additional experiments have shown that the number of constructed secondary candidates has decreased by 9% in the two methods.

6. CONCLUSION

In this paper, we have presented a novel method to improve the selection of DCP for dependent views in 3D-HEVC. A DV candidate has been inserted in the Merge candidate list in order to reduce the rate required for DCP, hence favoring its selection. Two insertion methods have been proposed, one where the DV is inserted in the secondary candidate list and another where the DV is inserted in the primary list. Bitrate reductions of -0.5% (-0.6% for method 2) and -0.6% for the two side views, along with -0.2% for synthesized and -0.2% for coded and synthesized views were reported. These reductions were accompanied by a 3% (4% for method 2) encoder runtime reduction since secondary candidates are less required to be constructed. Both methods were presented at the 2nd JCT3V meeting and method 2 was adopted in 3D-HEVC.

The gains obtained highly depend on the quality of the derived DV. The derivation process for our DV candidate is the same as the one used for the multiview candidate. Improving this process can achieve coding gains due to the improvement of the multiview candidate which depends on it, but also, due to the improvement of our newly added interview candidate.

Hence, it is an interesting topic for future work.

6. REFERENCES


