Modification of the disparity vector derivation process in 3D-HEVC

Elie Gabriel Mora #1, Joel Jung #2, Beatrice Pesquet-Popescu *3, Marco Cagnazzo *3

# Orange Labs
38 rue du G. Leclerc, 92794 Issy les Moulineaux, France
{1 elie.mora, 2 joelb.jung}@orange.com

** Telecom ParisTech, Signal Processing Department*
46 rue Barrault, 75634 Paris, France
{3 pesquet, 4 cagnazzo}@telecom-paristech.fr

Abstract—**The up-and-coming extension of HEVC for 3D video (3D-HEVC) includes various tools to exploit different redundancies in a 3D video signal. Inter-view redundancies are in particular exploited using Inter-View Motion Prediction (IVMP) and Inter-View Residual Prediction (IVRP). Both of these tools compensate disparity-wise the current prediction unit (PU) in order to find its corresponding PU in a base view, from which some prediction information for the current PU is retrieved. The disparity vector (DV) used for disparity compensation is currently derived using a neighboring search process (NBDV) for a DV across spatial and temporal neighbors. The first DV found is selected as the final DV used in IVMP and IVRP, with no guarantee of optimality.**

In this paper, the NBDV derivation process is changed: all found DVs from different neighbors are stored in a list. Redundant vectors in this list are removed, and a median computation on the remaining vectors is performed. The resulting DV is set as the DV used for IVMP. Average bitrate reductions of 0.6% and 0.8% for the two dependent views and 0.2% on synthesized views are reported with only a slight increase in encoder and decoder runtimes.

I. Introduction

Standardization activities are recently focusing on the development of a 3D video extension of HEVC called 3D-HEVC [1], following an initial response to a Call for Proposals [2] in 2011, and the formation of a joint collaborative team on 3D video, called JCT-3V, between ITU/T VCEG and ISO/IEC MPEG in 2012. 3D-HEVC exploits spatial, temporal, inter-component and inter-view dependencies in order to efficiently code the 3D information.

Inter-view redundancies between a currently coded dependent view and a previously coded base view which serves as a reference, are in particular exploited using the Inter-View Motion Prediction (IVMP) and the Inter-View Residual Prediction (IVRP) coding tools [1]. In both methods, a currently coded prediction unit (PU) in a dependent view is compensated, in the view axis, using a disparity vector (DV) in order to find its corresponding PU in the base view.

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00 ©2013 IEEE.

The Merge coding mode [3], initially introduced in HEVC, allows a PU to inherit the motion parameters (motion vectors + reference indices) of a neighboring PU. A candidate list, composed of the motion parameters of four spatial neighbors and one temporal neighbor, is formed and and the index of the most coding efficient candidate in the list is sent in the bitstream with an optional PU residual.

In 3D-HEVC, IVMP expands the Merge candidate list in the view axis by adding a multiview neighbor. Indeed, after finding the base PU using a derived DV, its motion vector, if it exists, is inserted in the first position of the Merge candidate list of the current PU. This candidate is commonly referred to as the multi-view candidate. The DV used to find the base PU is also inserted as an inter-view candidate in the fourth position [4] of the list. If the base MV does not exist (the base PU is coded in Intra mode for instance), the DV is set as the multi-view candidate and the inter-view candidate does not exist. The base MV and the DV are also inserted in the AMVP candidate list [3] where they are used as predictors for, respectively, the MV or DV of the current PU. A vector residual is thus transmitted in this case. In IVRP, the residual samples of the base PU are used to predict the residual samples of the current PU in order to further reduce the residual energy for more efficient compression.

The DV used for disparity compensation in IVMP and IVRP can be estimated. Multiple disparity estimation techniques ranging from classic block matching algorithms to more advanced stereo matching methods [5] or convex optimization approaches under illumination variations [6] can be used. However estimating the DV necessarily implies transmitting it in the bitstream for decodability. To avoid this costly transmission, the DV can be derived using already coded information. Specifically, in the current 3D-HEVC draft, it is derived using a search process for a DV across spatio-temporal neighboring positions. This neighboring position setup has been used until now in video coders for deriving a MV predictor [7], or a MV for direct inheritance [3].

The first DV found in this process, called Neighbor Disparity Vector (NBDV), is selected with no guarantee of optimality. Various methods have been proposed to improve NBDV, but none dealt with the sub-optimality induced by selecting the first DV found in the search process. In this paper, we propose a solution to this problem with a method that first stores the DVs of all the checked neighbors in a single list. Second, the redundant vectors are removed from this list, and finally, the median of the remaining vectors is computed and is set as the final DV which will be used for IVMP (for IVRP, the method is not applied, the first found DV is selected).

The rest of this paper is organized as follows: Section II presents the state of the art in DV derivation processes for 3D-HEVC. Section III describes the proposed method and its variants. The corresponding results are presented in Section IV with a detailed interpretation. Section V concludes this paper while underlining possibilities for future work.

II. State of the art

In this section, we will detail the different DV derivation processes used in 3D-HEVC. We will present in particular the currently used NBDV method which we improve in this work.

In 3D-HEVC, the texture component is coded before the depth component. Consequently, when coding a PU in a dependent view, the DV pointing to the corresponding PU in the base view cannot be computed from the depth component because it has not been coded yet. Getting the DV from the original depth map would require transmitting it in the bitstream because otherwise, the process cannot be repeated at the decoder. In order to avoid this costly transmission, a depth map estimate is computed and maintained for each view using already coded texture information. The maximum depth value contained in the collocated PU in the depth map estimate is transformed into the required DV. This derivation process is called Depth Map Disparity Vector (DMDV) [1]. To obtain the depth map estimates, the coded disparity vector field between the first dependent view and the base view is transformed into a depth map which is then warped to the base view and to other dependent views. Over time, the estimated depth maps are motion-compensated using the same motion vector field as in texture and corrected with coded disparity vector fields.

The complex warpings and successive motion compensations that DMDV involves led to the development of a lighter, less complex derivation process: NBDV [8]. NBDV is a simple search process across neighboring positions. The PUs covered by these positions are checked if they are coded using disparity-compensated prediction (DCP), in which case they have a DV, and the first DV found is selected as the final DV used for IVRP and IVMP. The positions are depicted in Figure 1. There are 5 spatial positions denoted by A_1 (left), B_1 (above), B_0 (above-right), A_0 (below-left)and B_2 (above-left), checked in this order. A PU covered by one of these positions can have up to two vectors, one from each reference list (list 0 and list 1) and they are both checked if they are DVs. Temporal positions are checked next, and they consist of the center of the collocated PU (CTR) and collocated bottom-right PU (BR) in a maximum of 2 temporal reference pictures. Furthermore, if a neighboring spatial PU was coded in motion-compensated prediction (*i.e*, the PU has a MV in a specific reference list, not a DV), its MV could have been computed using IVMP which necessarily involves the derivation of a DV. Indeed, the MV could have been inherited from the multi-view candidate in Merge mode or predicted using the multi-view MV predictor in AMVP. Constructing the multi-view candidate in both lists (Merge & AMVP) requires a prior derivation of a DV, which would then be linked to the current MV. These special DVs, called DDVs [9], are also checked after the temporal neighbors in the following order A_0, A_1, B_0, B_1, B_2. Compared to DMDV, NBDV brings small losses (0.1% on coded+synthesized views) while reducing encoder and decoder runtimes by 8% [10].

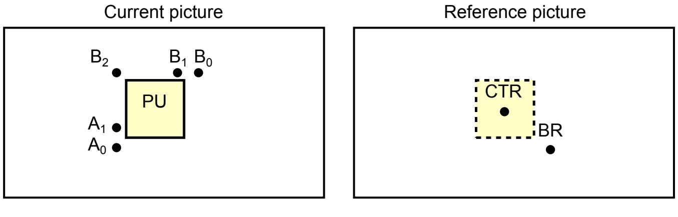


Fig. 1. Spatial and temporal neighboring positions in NBDV

Depth-oriented NBDV (DoNBDV) [11] is an interesting refinement of the classic NBDV process. It uses the coded depth map of the base view to refine the DV obtained after the standard NBDV process. Basicaly, the DV obtained is used to point to the corresponding PU in the base depth view. The maximum depth value inside that PU is converted into another DV which will be used for IVRP and IVMP. DoNBDV achieves significant bitrate reductions compared to NBDV (0.4% on coded views and 0.3% on coded+synthesized views) but adds a non-negligible decoding dependency between the base depth view and the dependent texture view (indeed, if the base depth view is corrupted, the dependent texture view cannot be decoded).

A final DV derivation process can be conceived if the depth is coded before the texture component. This is possible using the flexible coding order (FCO) tool which allows to change the coding order in 3D-HEVC. In this case, the DV can simply be computed from the coded depth component (taking the maximum depth value in the collocated depth PU and transforming it into a disparity) without the need of transmitting it since the process can be repeated at the decoder.

The DV derivation process in 3D-HEVC is subject to intensive research and is expected to change over the course of the standardization phase. At the time of writing this paper, and following the 2nd JCT-3V meeting, the derivation process currently used in 3D-HEVC is NBDV since it is coding efficient, not complex, and does not introduce new decoding dependencies. However, NBDV is sub-optimal. Indeed, the first DV / DDV found in a neighboring PU during the NBDV search process is selected as the final DV and the search process stops. The remaining neighbors are not checked even if some have a DV / DDV which is better, rate-distortion (R-D) wise, than the selected one hence the sub-optimality of the process. The proposed method answers and solves this specific issue.

III. Proposed method

A. Preliminary study

Table I shows the percentage of PUs coded in Merge mode using either the multi-view or the inter-view candidate in version 5.0.1 of the 3D-HEVC reference software: HTM, averaged across four QPs, for various tested sequences. The test conditions used are the same as the ones used to evaluate our method, which are described in Section IV-A. We can see that the multi-view Merge candidate is largely selected in HTM-5.0.1 (for 57% of PUs coded in Merge mode, on average) since it is inserted at the first position in the list (the rate needed to code a merge index of 0 is small, hence the R-D cost of this candidate is small as well). The inter-view candidate is inserted further down the list and is thus selected less often (only 1%).

Sequence	Multi-view	Inter-view
Kendo	51.6	2.1
Newspaper	53.4	1.4
Balloons	59.9	1.6
Dancer	45.9	1.7
GT Fly	65.6	0.9
Poznan Hall2	61.5	0.9
Poznan Street	60.6	1.2
Average	**56.9**	**1.4**

TABLE I
PERCENTAGE OF PUS CODED IN MERGE MODE USING THE MULTI-VIEW OR THE INTER-VIEW CANDIDATES

The efficiency of the multi-view and the inter-view candidate directly depends on the derived DV. If the DV quality is improved, the distortion associated to these two candidates will decrease, along with their R-D cost, hence increasing their selection and achieving coding gains. These gains will be significant since on the one hand, we are in general improving the Merge coding mode which is already widely selected (our experiments have shown that it is selected for 92% of PUs in the same test conditions), and on the other hand, we are improving the first candidate in the Merge list which is also largely selected as shown in Table I. It is important to note that some of our gains will also come from improving these candidates in the AMVP list but those gains are small compared to the ones resulting from the improvement in the Merge list.

B. Method description

In our method, the search process is never stopped. All the spatial and temporal neighbors are checked, in the usual order, and all found DVs and DDVs are stored together in a single list. A redundancy check is applied to remove redundant vectors in this list. Then, the median of all remaining vectors is computed and is set as the final DV used for IVMP. Applying the method for IVRP as well will be tested seperately in a variant. Figure 2 illustrates the different steps of our algorithm.

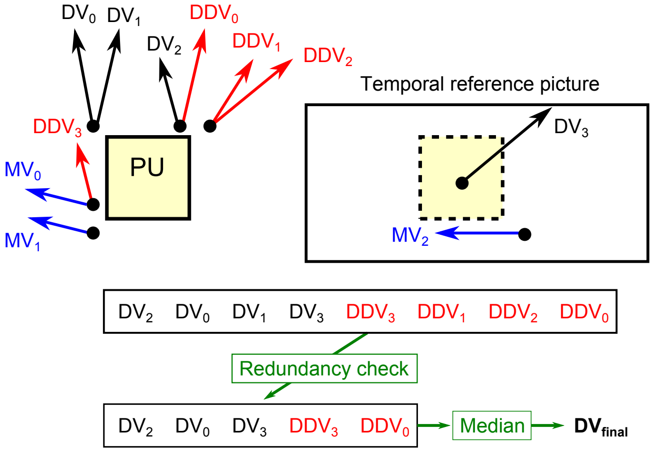

DV_0
DV_1
DDV_0
DV_2
DDV_1
DDV_2
DDV_3
MV_0
MV_1
PU
Temporal reference picture
DV_3
MV_2
DV_2 DV_0 DV_1 DV_3 DDV_3 DDV_1 DDV_2 DDV_0
Redundancy check
DV_2 DV_0 DV_3 DDV_3 DDV_0
Median
DV_{final}

Fig. 2. Proposed DV derivation method

The advantage of our method is that it groups different types of DVs, namely DVs obtained from DCP-coded PUs and DDVs obtained from MCP-coded PUs, in a single list. This heterogeneity in lists is usually coding efficient. For instance, in the Merge candidate list, secondary candidates are constructed to fill the list if primary candidates are unavailable. Hence, two types of candidates can potentially be in the same list, namely primary and secondary candidates. This configuration has been proven to bring coding gains compared to one which does not involve secondary candidates. Our method is thus set in the same mind frame.

The disadvantage of our method lies in the worst case scenario for median computation. Indeed, each spatial neighbor can have up to two vectors, one from each reference list, and there are 5 neighbors. In addition, there are two temporal neighbors in two temporal reference frames, each one having at most one vector. In case all spatial neighbors in both reference lists and all temporal neighbors have DVs or DDVs, and there is no redundancy between these vectors, the median has thus to be computed on 14 vectors. This is quite complex to perform in hardware. Consequently, in order to avoid this worst-case scenario, different variants of the method have been implemented and tested.

C. Variants

The 1REF variant consists of storing in the list a maximum of one vector per spatial neighbor. In case the spatial neighbor has two DVs or two DDVs, only the one from reference list 0 is stored. In case it has one DV and one DDV, only the DV is stored. In this configuration, the maximum number of spatial candidates is 5, making the worst-case maximum number of vectors on which the median is computed (*MaxCand*) equal to 9. Another variant, RMPOS, consists in simply removing one or more spatial positions (for example A_0) from the check, hence decreasing *MaxCand* by 2 (or by 1 if associated with 1REF) for each spatial position removed. In our experiments, RMPOS consisted in removing the A_0 and the B_2 spatial positions from the check. A final variant aimed at reducing *MaxCand*, called LIMIT-X, consists in storing only the first X found DVs / DDVs in the list. In this case, *MaxCand* =

X. Note that the LIMIT-1 variant is equivalent to the standard NBDV process.

The following variants are not aimed at reducing *MaxCand*, but rather implemented and tested to make interesting interpretations: NODDV does not store any DDVs in the list, ALLOWRED removes the redundancy check before median computation, NOAMVP does not apply our method for AMVP while APPLYRES applies it for IVRP as well, and finally, MEAN replaces the median computation with the computation of the vectors' average.

IV. Experimental results

A. Experimental setting

We have implemented the proposed method and all of its variants in HTM-5.0.1 [12]. We have strictly followed all the common test conditions (CTCs) defined by JCT-3V [13]. 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 method and the variants 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 three texture and three depth views (one central base view and two side views). After encoding, three intermediate views are synthesized between the left and the center view, and another three between the center and the right views. The renderer used is the one included in the HTM-5.0.1 package. This renderer, called "VSRS-1D-Fast", interpolates an intermediate view from a left and right reference. Remaining holes due to disocclusions are filled using a line-wise inpainting. PSNRs on synthesized views are computed with respect to synthesized views rendered with uncompressed original texture and depth views. Coding gains are measured with the Bjontegaard delta (BD-Rate) metric [14].

B. Coding gains

1) Objective results: Table II gives the coding gains (negative values are gains) achieved with our method. The anchor considered is HTM-5.0.1 under the same common test conditions. The results of our method are summarized also in Table III along side all the studied variants (only the average results accross all sequences are given in Table III). 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 "Synt." column gives results on the six synthesized views (the bitrate considered is the sum of the three texture and depth bitrates, and the PSNR is the average PSNR of all six synthesized views). The "Coded+Synt." result is the same as in the previous column except that the PSNR considered is the average PSNR of the six synthesized views and the three coded texture views. In Table III, an additional column, "MaxCand" was added to show the maximum number of vectors on which the median can be computed in a worst-case scenario, per variant. Note that there is a ±3% error margin on the encoder and decoder runtimes, because even if launched back to back on the same machine, the runtime of an encoding or decoding process varies only slightly each time.

Sequence	**Video**				**Synt.**	**Coded +Synt**	**Runtimes**	
	0	**1**	**2**	**Avg**			**Enc**	**Dec**
Balloons	0.0	-0.7	-0.7	-0.3	-0.2	-0.2	100	99
Kendo	0.0	-1.2	-1.3	-0.5	-0.4	-0.4	98	97
Newspaper	0.0	-0.7	-0.7	-0.3	-0.2	-0.2	99	98
GT Fly	0.0	-0.6	-0.9	-0.2	-0.2	-0.2	89	98
Poznan Hall2	0.0	0.0	-0.5	-0.1	-0.2	-0.2	102	101
Poznan Street	0.0	-0.4	-0.4	-0.1	-0.1	-0.1	104	95
Dancer	0.0	-0.9	-1.0	-0.3	-0.4	-0.4	108	99
Average	**0.0**	**-0.6**	**-0.8**	**-0.3**	**-0.2**	**-0.2**	**100**	**98**

TABLE II
BD-Rate coding results per sequence, in %, with the proposed method

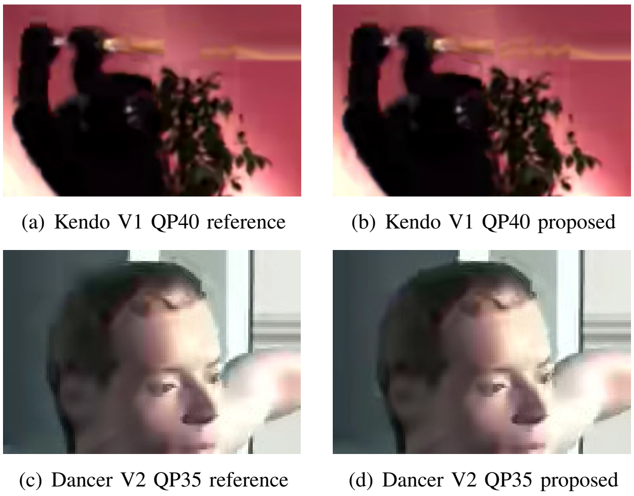
Variant	**Max Cand**	**Video**				**Synt.**	**Coded +Synt**	**Runtimes**	
		0	**1**	**2**	**Avg**			**Enc**	**Dec**
Method	14	0.0	-0.6	-0.8	-0.3	-0.2	-0.2	100	99
1REF	9	0.0	-0.6	-0.7	-0.2	-0.2	-0.2	97	98
1REF+RMPOS	7	0.0	-0.5	-0.6	-0.2	-0.2	-0.2	99	99
LIMIT-4	4	0.0	-0.5	-0.7	-0.2	-0.2	-0.2	103	98
NODDV	14	0.0	-0.3	-0.3	-0.1	-0.1	-0.1	97	99
ALLOWRED	14	0.0	-0.2	-0.3	-0.1	-0.1	-0.1	98	98
MEAN	14	0.0	-0.3	+0.1	0.0	0.0	0.0	100	98
NOAMVP	14	0.0	-0.6	-0.7	-0.3	-0.2	-0.2	101	98
APPLYRES	14	0.0	-0.6	-0.8	-0.3	-0.2	-0.2	108	98

TABLE III
Average BD-Rate coding results for the different variants

Table II shows 0.6% and 0.8% average bitrate reductions on the dependent views, and 0.2% on synthesized views, with a *MaxCand* of 14, as explained in Section III-B. These gains were achieved with no increase on encoder or decoder runtimes. Note that no gains are reported on the central view because our method is simply not applied there (no DV derivation is done on the base view).

Table III shows the average coding results of three variants (1REF, 1REF+RMPOS, LIMIT-4) aimed at reducing *MaxCand*. These variants slightly reduce the gains obtained in the original method but alleviate the median computation in hardware in the worst-case scenario. The NODDV, ALLOWRED and MEAN variants however keep the same *MaxCand* as in the original method but significantly reduce the gains (losses are even reported for the MEAN variant on the second dependent view). Finally, the NOAMVP variant slightly reduces the gains while not affecting runtime, while the APPLYRES variant, on the contrary, achieves the same coding performance as the original method with the same *MaxCand* but with an increase in encoder runtime (108%).

2) Visual results: The significant gains on the dependent views for the Kendo and Dancer sequences in the proposed method are visible in Figure 3. Parts of the left view (view 1) and the right view (view 2) at a QP of 40 and 35 for the Kendo and Dancer sequences respectively, coded using the HTM-5.0.1 reference software and with the proposed method are shown in this figure. For the Kendo sequence, we can see that our method avoids having the sword broken in two as in the reference. For the Dancer sequence, the back of the dancer's head is more sharply represented using our method.


(a) Kendo V1 QP40 reference

(b) Kendo V1 QP40 proposed

(c) Dancer V2 QP35 reference

(d) Dancer V2 QP35 proposed

Fig. 3. Parts of dependent views coded with the reference software and with the proposed method

C. Results interpretation

1) Origin of the gains: The proposed method improves the quality of the DV used in IVMP. Consequently, the multi-view and the inter-view candidates in the Merge list, which depend on that DV, are also improved and more often selected. Table IV shows the increase in the number of PUs coded in Merge mode using the multi-view or the inter-view candidates, in the proposed method, for each tested sequence, averaged across four QPs. A significant increase is noted for the inter-view candidate (31% on average) since it directly corresponds to the improved DV. For the multi-view candidate, the improved DV is only used to find a PU in the base view from which to extract a MV. Consequently, the improved DV may point to a PU that has the same MV as the one of the PU pointed to by the original DV. In this case, our DV improvement has no effect, and this explains why on average, the selection of the multi-view candidate has only slightly increased (2%). In any case, these increases are directly correlated with the coding gains achieved using our method.

2) Runtime results analysis: Furthermore, Table V shows the average, minimum and maximum number of vectors on which the median is computed for each tested sequence in the encoder and the decoder, in the proposed method. We can see that the worst case scenario in which the median is computed on 14 values never occurs for any sequence (maximum is 12). On average, the median is computed on 1.9 vectors at the encoder and 2.2 vectors at the decoder, the difference being due to the fact that the encoder tests all possible CU sizes and partitions and hence performs the median computation much more often than the decoder. In any case, most of the time, the median computation is simple and is performed quickly. This explains why the runtime increase at both coder sides was imperceptible. Indeed, this increase definetely exists since our method necessarily adds some operations to the encoder and decoder without removing others, but it is not visible in Table II because it is really small.

Sequence	Multi-view increase	Inter-view increase
Kendo	0.6	23.3
Newspaper	1.4	21.5
Balloons	3.6	18.4
Dancer	3.5	54.8
GT Fly	0.5	62.9
Poznan Hall2	2.2	16.7
Poznan Street	1.4	20.1
Average	**1.9**	**31.1**

TABLE IV
INCREASE IN THE PERCENTAGE OF PUS CODED IN MERGE MODE USING THE MULTI-VIEW OR THE INTER-VIEW CANDIDATES

Sequence	Encoder			Decoder		
	Avg	Min	Max	Avg	Min	Max
Kendo	1.9	1	11	2.2	1	9
Newspaper	1.9	1	11	2.1	1	10
Balloons	1.9	1	10	2.2	1	10
Dancer	1.9	1	10	2.2	1	10
GT Fly	2.3	1	12	2.4	1	12
Poznan Hall2	1.7	1	11	1.9	1	10
Poznan Street	2.0	1	11	2.2	1	11
Overall	**1.9**	**1**	**12**	**2.2**	**1**	**12**

TABLE V
AVERAGE, MINIMUM AND MAXIMUM NUMBER OF VECTORS FOR MEDIAN COMPUTATION AT THE ENCODER AND DECODER SIDE

3) Variants results interpretation: The 1REF, 1REF+RMPOS and the LIMIT-4 variants all succeed in reducing *MaxCand* with a small penalty on coding gains (0.1% on coded texture videos). The performance of these three variants is roughly equivalent, but LIMIT-4 reduces *MaxCand* the most (to 4 instead of 9 or 7), making it clearly the best variant in this category. Note that Table III shows that LIMIT-4 increases the encoder runtime (103%) but as previously said, there is a ±3% error margin on this runtime so any increase below 103% or any decrease above 97% is not considered valid.

The gains are more significantly reduced in the ALLOWRED variant, in which the redundancy check on the vectors before median computation is not performed. This can be explained by the fact that the redundancy check allows to diversify the input vectors for the median computation, hence avoiding having the same DV chosen over a contiguous region with different disparity values. In addition, the redundancy check reduces the average and maximum number of vectors (considered on all sequences) on which the median is computed. Indeed, our experiments show that these values would have increased to 4.0 and 14 at the encoder, and 4.8 and 14 at the decoder, respectively, if the check was not performed.

Storing only the DVs in the list while discarding DDVs also reduces the gains of our method. Indeed, not considering DDVs in the list penalizes our method in case there are no DVs to insert. Indeed, in that case, the final DV used for IVMP is set to the zero vector, while in the reference method, a DDV may be chosen, which is almost always more accurate than the zero vector. This result also validates our intuition discussed in Section III-B about the fact that the heterogeneity in lists in a video coder is more efficient than homogeneity.

If we do not apply our method for AMVP, the multi-view and the inter-view candidates in the AMVP list are not improved. Consequently, a slight reduction of the gains on the dependent views (0.1% loss) is noted with practically no influence on encoder runtime. This validates our assumption that the contribution of improving the multi-view and inter-view AMVP candidates in the proposed method is small.

If our method is applied for IVRP as well as IVMP, as in the APPLYRES variant, the coding gains would remain the same on average. This is because improving the multi-view and the inter-view Merge candidates has a much higher impact than improving IVRP. However, the slight increase in encoder runtime in our method becomes multiplied by around 2.5 since IVRP is applied for all PUs, including PUs coded in Intra, as opposed to IVMP. As a consequence, it becomes visible as seen in Table III. Hence, for a better coding efficiency / complexity tradeoff, our method should not be applied for IVRP.

Finally, we have tested replacing the median computation with a simpler average computation (the MEAN variant). However, the coding gains obtained are small. Some losses are even reported for the second dependent view. This can be explained by the fact that the median allows to select a DV out of accurate, previously estimated DVs, whereas the average creates a new DV which might not truly describe the disparity at the level of the current PU.

V. Conclusion and future work

In this paper, we have presented a method that tackles the sub-optimality problem in the current DV derivation process in 3D-HEVC (NBDV) resulting from selecting the first DV or DDV found in the search. In our method, all found DVs and DDVs in spatial and temporal neighboring PUs are stored together in a single list, and the search process is never stopped. Redundant vectors in the list are removed, and the median of the remaining vectors is computed and set as the final DV used for IVMP. Average bitrate reductions of 0.6% and 0.8% on the two dependent views, along with 0.2% on synthesized views were achieved with no increase in encoder and decoder runtimes. Several variants were tested as well, in order to either reduce the worst-case maximum number of vectors on which the median is computed, or to provide informative results.

The selection of the final DV can also be based on an R-D check applied on the candidates stored in the list. The DV selected would be the one yielding the lowest R-D cost. This requires sending the index of the DV in the list to the decoder, but the method might still bring significant gains. Hence, it is an interesting idea to consider for future work.

Acknowledgment

The authors acknowledge the support of the Persee project (ANR-09-BLAN-0170)

References

[1] G. Tech, K. Wegner, Y. Chen, and S. Yea, “3D-HEVC test model 2,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-B1005, October 2012.

[2] “Call for proposals on 3D video coding technology,” ISO/IEC JTC1/SC29/WG11 N12036, March 2011.

[3] P. Helle, S. Oudin, B. Bross, D. Marpe, M. Bici, K. Ugur, J. Jung, G. Clare, and T. Wiegand, “Block merging for quadtree-based partitioning in HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1720 –1731, December 2012.

[4] E. Mora, J. Jung, B. Pesquet-Popescu, and M. Cagnazzo, “3D-CE5.h related: Modification of the merge candidate list for dependant views in 3DV-HTM,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-B0069, October 2012.

[5] D. Scharstein, R. Szeliski, and R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” in *IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*, 2001, pp. 131–140.

[6] W. Miled, J. Pesquet, and M. Parent, “A convex optimization approach for depth estimation under illumination variation,” *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 813–830, April 2009.

[7] G. Laroche, J. Jung, and B. Pesquet-Popescu, “RD optimized coding for motion vector predictor selection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1247–1257, September 2008.

[8] L. Zhang, Y. Chen, and M. Karczewicz, “3D-CE5.h related: Disparity vector derivation for multiview video and 3DV,” ISO/IEC JTC1/SC29/WG11 MPEG2012/m24937, April 2012.

[9] J. Sung, M. Koo, and S. Yea, “3D-CE5.h: Simplification of disparity vector derivation for HEVC-based 3D video coding,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-A0126, July 2012.

[10] L. Zhang, Y. Chen, and M. Karczewicz, “CE5.h: Disparity vector generation results,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-A0097, July 2012.

[11] Y.-L. Chang, C.-L. Wu, Y.-P. Tsai, and S. Lei, “CE5.h related: Depth-oriented Neighboring Block Disparity Vector (DoNBDV) with virtual depth retrieval,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-B0090, October 2012.

[12] G. Tech, “HTM-5.0.1 software,” Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-5.0.1.

[13] D. Rusanovsky, K. Muller, and A. Vetro, “Common Test Conditions of 3DV Core Experiments,” ITU-T SG 16 WP 3 & ISO/IEC JTC 1/SC 29/WG 11 JCT3V-B1100, October 2012.

[14] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, Austin, USA, April 2001.