

BUFFER REQUIREMENT ANALYSES FOR MULTIVIEW VIDEO CODING

Ying Chen¹, Ye-Kui Wang², Moncef Gabbouj¹

¹Institute of Signal Processing, Tampere University of Technology

²Nokia Research Center

ABSTRACT

Multiview video coding (MVC), which is becoming an extension of H.264/AVC, is currently under development by the Joint Video Team (JVT). Compared to H.264/AVC, the main new compression tool in MVC is inter-view prediction, which, among others, causes a substantial increase of the decoded picture buffer (DPB) size. Therefore to have an efficient buffer management for MVC is highly desirable. In this paper, we provide analyses of minimum buffer requirements for typical MVC coding structure with two coding methods, view-first coding and time-first coding. The analysis results are helpful in designing reference picture management or reference picture marking methods.

Index Terms— Multiview video coding, decoded picture buffer, reference picture marking, H.264/AVC

1. INTRODUCTION

Multiview video technologies have gained significant interest recently. Two typical applications are free-viewpoint video and 3D TV. In free-viewpoint video, the viewer can interactively choose his/her viewpoint in 3-D space to observe a real-world scene from preferred perspectives [1]. In 3D TV, there are different stereoscopic views generated by the video captured by multiple cameras of the scene. Comparing to free-viewpoint video, 3D TV does not require interaction [2]. On the other hand, 3D TV usually requires displaying of all the views; while free-viewpoint TV displays only one view. Due to the huge amount of data, particularly when the number of views to decode is large, the transmission part of the system for multiview video applications relies heavily on the compression of the video captured by cameras.

Simulcast coding can be employed to multiview video coding using one video coder, e.g., H.264/AVC standard [3] for each view separately. However, exploiting of the correlation for further improved compression efficiency is of great interest. Fortunately, inter-view prediction is supported in the latest draft specification of multi-view coding extension of H.264/AVC (MVC), which was decided by MPEG to be a start point for MVC after the subjective assessment among several other codecs.

The latest draft of the video model of MVC is described in JMVM [4]. In MVC, a picture can use

pictures of different views with the same time instance for inter-view prediction reference. For each view, the information of which views may be used for inter-view prediction reference is included in the extension to the sequence parameter set (SPS). This information stays unchanged throughout a coded video sequence associated with the SPS.

Temporal scalability supported by H.264/AVC [3], is inherited in MVC. The most typical coding structure for temporal scalability is the hierarchical B picture coding structure [5]. Typically, the hierarchical structure requires larger DPB size compared to the simple structures such as IPPP and IBBP. The two main H.264/AVC DPB management tools, reference picture list reordering (RPLR) and memory management control operation (MMCO) commands, are typically utilized in hierarchical B coding.

In H.264/AVC, reference pictures are marked as short-term or long-term pictures. There are two types of operations for the reference picture marking: adaptive memory control and sliding window. Different reference picture marking operations can be applied to each picture independently. The adaptive memory control can explicitly mark a short-term or long-term picture as “unused for reference”, while the sliding window operation is a first-in-first-out mechanism among short-term reference pictures.

Because in MVC more than one view is encoded and inter-view prediction is employed, the required DPB size for decoding an MVC bitstream could be very large, as can be seen from the buffer requirement analyses presented later in this paper. Therefore, to design an optimal buffer management for pictures both used for prediction reference and waiting for output with the considerations of coding order, temporal scalability and view scalability is crucial for the memory resource control.

In this paper, we present DPB analyses for minimum buffer requirements for the most typical coding structure included in JMVM [4] with two different coding methods, time-first coding and view-first coding. The prediction structure is represented as a binary tree to ease the analyses. The analysis results are helpful in designing reference picture management method, in particularly, reference picture marking method for multiview video coding. For example, the authors have utilized the results in their MVC reference picture marking proposal [6], which has been adopted into the Joint Draft of MVC [7].

2. MVC PREDICTION STRUCTURES AND DECODING ORDERS

A typical prediction structure (including both inter prediction within each view and inter-view prediction) is shown in Fig.1 [8], where a time instance is denoted as T_m and a view is denoted as S_n , and where predictions are indicated by arrows, the pointed-to object uses the point-from object for prediction reference. In MVC, anchor pictures are defined. Anchor pictures can be used as random access points, such that all the subsequent pictures in display order can be correctly decoded after the random access. Except for the very first group of picture (GOP), i.e. the pictures of T_0 in Fig.1, which only includes the anchor pictures, typically other GOPs, e.g., the pictures of T_1 to T_8 , include both non-anchor and anchor pictures. Each GOP includes n_v (number of views) views and in each view gl (GOP length) pictures. For example, in Fig. 1, n_v is 8 and gl is 8.

Within a GOP, if pictures of the same view are contiguous in decoding order, the coding method is referred to as view-first coding. If pictures of the same time instance are contiguous in decoding order, the coding method is referred to as time-first coding. In other words, in view-first coding, the decoder would not start decoding another view until all the pictures of the current GOP in the previous view are decoded; while in time-first coding, the decoder would not start coding pictures of another time instance until the pictures in the current time instance are all decoded. It should be noted that both coding methods supports any prediction structures, e.g. the one shown in Fig.1. The most essential difference between the time-first coding and view-first coding lies in the DPB size requirement.

Time-first coding is mandated in the latest MVC specification [8].

3. BUFFER REQUIREMENT FOR TEMPORAL SCALABILITY IN SINGLE VIEW VIDEO CODING

In hierarchical B picture coding for a single view, a GOP usually includes a key (anchor) picture which is coded as I or P frame and other pictures which are coded as B pictures, as shown by the pictures of view 0 (denoted as s_0) in Fig.1. The B pictures are predicted hierarchically based on the display order, which corresponds to picture order count (POC) specified in H.264/AVC. Each B picture is predicted from the lower temporal level pictures that are closest in display order in both forward and backward directions. Coding of a GOP needs only the key pictures of the previous GOP besides those pictures in the current GOP.

Let the relative POC value, denoted as $POC_{IdInGOP}$, be equal to the POC value of the current picture minus the POC value of the anchor picture in the previous GOP. Note that any integer n can be factorized as the x power of 2 multiple an odd integer y , i.e., $n = 2^x y$, where x is an integer. When the value $POC_{IdInGOP}$ is represented as

$POC_{IdInGOP} = 2^x y$, x is then equal to the difference between TL and the temporal level value of the picture, where $TL = \log_2(gl)$.

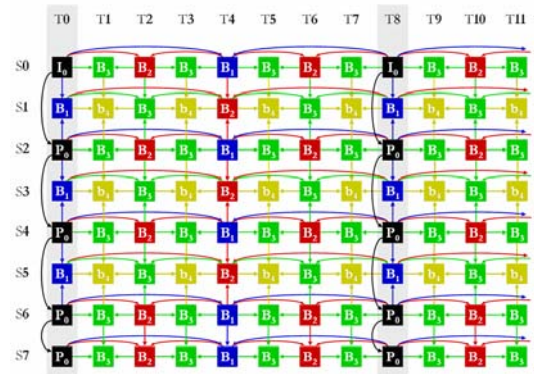


Fig. 1. Typical MVC prediction structure

As shown in Fig 2, the prediction structure of non-anchor pictures can be represented as a binary tree, where each node represents a picture and can have its ancestor nodes as reference pictures. A picture does not refer to a picture with higher or equal temporal level, which enables discarding of higher level pictures for temporal scalability. The left-most node in each temporal level has y equal to 1.

Only pictures with the highest temporal level, equal to TL , are coded as non-reference pictures

With the binary tree structure, we can see that the minimum DPB size required for hierarchical B coding structure equals to the number of nodes passed when the first temporal level 0 node is to be traveled by depth first order, i.e. $TL+1$. For the example shown in Fig.2, the required DPB size is 5 decoded pictures. After node 2 is coded or during coding of node 3, the DPB reaches its maximum size. Later, once a node corresponding to a reference picture is being coded, it is needed to invoke MMCO commands and the decoded picture removal process to remove the previous node P which belongs to the same temporal level (as the current coding node). For example, after node 6 is coded, node 2 can be removed. After an anchor picture is coded, the previous GOP pictures are to be removed except for the previous anchor picture.

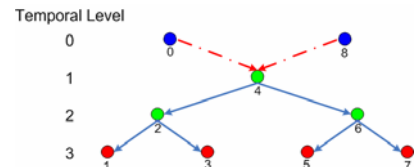


Fig. 2. Prediction structure of non-anchor pictures represented as a binary tree

4. BUFFER REQUIREMENT FOR MULTIVIEW VIDEO CODING

In the typical MVC prediction structure as shown in Fig.1, when an odd view is being coded, pictures belonging to

its neighboring views are required for inter-view prediction reference. The minimum DPB sizes required for both view-first coding and time-first coding are presented in this section.

4.1. View-first coding

To consider temporal scalability, a reference picture can only be removed by another picture that has the same or lower temporal level. For MVC, to consider view scalability, i.e. views in the ending positions of the view dependency path can be discarded, a reference picture can only be removed by another picture in the same time instance that has the same or lower view level. In the typical example as shown in Fig 1, because of view dependencies, a picture in an odd view will never remove a picture in an even view. We consider the status when view 1 is being coded. The DPB status is derived according to the following steps.

1. All the nv anchor pictures from the previous GOP will be stored in the DPB.
2. All the $(2 \cdot gl)$ pictures of view 0 and view 2 will be stored in the DPB.
3. Coding pictures in view 1 using the hierarchical B structure. With the same analysis as Section 3, TL additional pictures will be stored into the DPB. Note that the anchor picture from the previous GOP has already been considered in step 1.

When view 4 is being coded, pictures in view 2 and view 1 are not needed anymore and can be removed. Therefore, the maximum DPB size is equal to $(nv + 2 \cdot gl + TL)$ decoded pictures. The DPB status when the maximum number of stored pictures is reached is shown in Fig 3. For the example shown in Fig.1, the minimum DPB size is 27 decoded pictures. The DPB size is 44 if the GOP length is changed to 16.

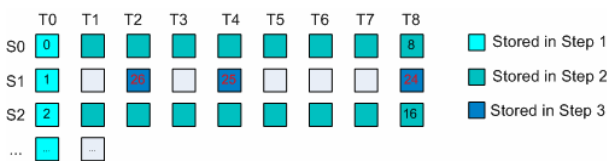


Fig. 3. DPB status for view-first coding

4.2. Time-first coding

In time-first coding, each view preserves the hierarchical B coding structure. When the pictures of T1 in Fig.1 are being coded, the DPB status is derived according to the following steps.

1. Pictures in all the views with temporal level or x greater than 0 (i.e. reference pictures) and y equal to 1 will be stored in the DPB. The numbers of the nodes are 0, 8, 4, and 2 as shown in Fig. 4. The number of these pictures is $nv \cdot (TL + 1)$.

2. When the picture T1/S1 (referring to Fig.1) is being coded, pictures S0/T1 and S2/T1 are used for inter-view reference and will be stored in the DPB.

For other pictures in time instance T1, the number of pictures that are used for inter-view reference is either less than or equal to 2.

If a picture in one time instance is coded and its temporal level is not the highest, the previous picture in decoding order with the same temporal level from the same view can be removed.

Therefore, the maximum DPB size is equal to $(nv \cdot (TL + 1) + 2)$ decoded pictures. The DPB status when the maximum number of stored pictures is reached is shown in Fig 4. For the example shown in Fig 1, the minimum DPB size is 34 decoded pictures. The DPB size is 44 if the GOP length is changed to 16.

Actually, the result is a generalization of the result for temporal scalability in AVC. Each node (or super node) now in the MVC case is one time instance. But this “super node” only refers to the time instance that contains pictures for temporal prediction.



Fig. 4. DPB status for time-first coding

4.3. Buffer analysis with output consideration

The above analyses do not consider pictures stored in the DPB waiting for output, which is taken into account in this subsection.

In AVC, for a non-reference picture, its output time may be later than the time it is decoded. For a reference picture, its output time may be later than the time when it is marked as “unused for reference”. Both these increase the required DPB size. Due to space limit, we skip details of the following analyses.

For the hierarchical B picture coding, to meet the maximum difference between the decoding order and the output order of any picture, picture output can only start after the picture with $POCIdInGOP$ equal to 1 is decoded. Consequently, compared to the minimum buffer size analyzed in previous subsections, which should be equal to num_ref_frames , the increased buffer size is $(TL - 1 - \log_2[TL - 1])$. This number also indicates the end-to-end delay of the coding structure, in units of picture period.

In MVC, simultaneous outputting of all the pictures in one view is required. For some applications, only a subset of the encoded views (M) are targeted for the output and they are dependent on other K views not for output, where $M + K \leq nv$.

For 3D TV, if view-first coding is used, all views are to be stored in the DPB until the last view (in decoding order) starts outputting. At that moment, the first highest

temporal level picture is decoded and the $TL+1$ pictures of the last view are stored in the DPB. After that, each time a picture of the same GOP is decoded, at least $nv-1$ pictures can be removed from the DPB. Therefore the DPB size required for 3D TV view-first coding is $(nv-1)gl+TL+1$. For time-first coding, the constraint between output time and marking time is the same for every view and therefore the required DPB size is $nv(2 \cdot TL - \log_2 \lfloor TL-1 \rfloor)$.

For free-viewpoint video, one view is required for the output. Let the number of directly dependent views of the desired view be $nddv$ and the number of related views (dependent on views and the desired view) be nrv . For view-first coding, usually the anchor pictures are stored in DPB, with a size of nrv . For the directly dependent views, all pictures need to be stored in DPB until the desired view starts decoding. The size is $nddv \cdot gl$. Then TL pictures of the desired view are added into the DPB before outputting. The DPB size decreases when the output starts. Thus, the DPB size for view-first coding is $nrv + nddv \cdot gl + TL$. Here, nrv is usually equal to the number of views and $nddv$ is equal to the maximum number of inter-view reference pictures. Note that the pictures in the dependent views are not removed during the decoding of pictures in the target view because usually they will be marked as "unused for reference" somewhere else, e.g., pictures of the $nddv$ views are to be removed using some implicit process after the target view is decoded.

For time-first coding, all pictures of the related nrv for those dependent on $nrv-1$ views, only DPB with a size of num_ref_frames shall be maintained. num_ref_frames is $TL+1$. For the target picture, a DPB of $2 \cdot TL - \log_2 \lfloor TL-1 \rfloor$ is required. Besides, $nddv$ highest temporal level pictures in other views are to be temporally stored in DPB. So the total DPB size for time-first coding is:

$$(nrv-1) \cdot (TL+1) + 2TL - \log_2 \lfloor TL-1 \rfloor + nddv = nrv \cdot (TL+1) + TL - 1 - \log_2 \lfloor TL-1 \rfloor + nddv$$

Obviously, for the most memory demanding case, $nddv$ is 2 and nrv is nv . The DPB size for view-first coding is $nv + 2 \cdot gl + TL$. For time-first coding, the DPB size is $(nv+1) \cdot (TL+1) - \log_2 \lfloor TL-1 \rfloor$.

Assume as a random variable, the number of required views obeys a uniform distribution, then expectations of nrv is $E(nrv) = nv/2$. Since $nddv$ can be 2 or 1 with same probabilities, we have $E(nddv) = 3/2$.

Therefore the average DPB sizes for free-viewpoint video for time-first and view-first are $(nv+3gl)/2+TL$ and $(nv \cdot (TL+1) + 3)/2 + TL - 1 - \log_2 \lfloor TL-1 \rfloor$, respectively.

Table 2 DPB size for MVC view-first coding and time-first coding in different scenarios.

	DPB (w/o output)	3D TV DPB (with output)	Free viewpoint video DPB (with output) Maximum	Free viewpoint video DPB (with output) Average
view-first	$nv + 2 \cdot gl + TL$	$(nv-1)gl + TL + 1$	$nv + 2 \cdot gl + TL$	$(nv + 3gl)/2 + TL$
time-first	$nv \cdot (TL+1) + 2$	$nv(2 \cdot TL - \log_2 \lfloor TL-1 \rfloor)$	$(nv+1) \cdot (TL+1) - \log_2 \lfloor TL-1 \rfloor$	$(nv \cdot (TL+1) + 3)/2 + TL - 1 - \log_2 \lfloor TL-1 \rfloor$

Table 1 gives a typical example of the DPB sizes for MVC with different coding scenarios while Table 2 provides analytic results.

5. CONCLUSIONS

Because of the hierarchical B picture coding structure applied for temporal scalability as well as the view dependencies and large number of views, multiview video coding is memory consuming. So it is curtail to design efficient coding structure to minimize the required decoded picture buffer. Based on our previous analysis, in the view dimension, the proposed time-first coding requires much less DPB size comparing to the view-first coding with the same coding efficiency. While in the temporal domain, marking of the pictures in the same temporal level is utilized. Therefore, we reach an optimal design for MVC in terms of memory requirement. The time-first coding now is mandatory for MVC specification.

6. REFERENCES

- [1] A. Smolic, and P. Kauff, "Interactive 3D Video Representation and Coding Technologies", *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery*, Jan. 2005
- [2] A. Vetro, W. Matusik, H. Pfister, J. Xin, "Coding Approaches for End-to-End 3D TV Systems," *Picture Coding Symposium*, 2004
- [3] Wiegand, T., Sullivan, G. J., Bjontegaard, G., Luthra, A., "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, Jul. 2003
- [4] "Joint Multiview Video Model (JMVM) 5.0," *JVT-X207*, Geneva, Switzerland, Jun.-Jul. 2007
- [5] H. Schwarz, D. Marpe, and T. Wiegand "analysis of hierarchical B pictures and MCTF," *Proc. ICME*, Toronto, Canada, 2006
- [6] Y. Chen, Y.-K. Wang, M. M. Hannuksela, "MVC Reference Picture Management," *JVT-U105*, Hangzhou, China, Oct. 2006
- [7] "Joint Draft 4.0 on Multiview Video Coding," *JVT-X209*, Geneva, Switzerland, Jun.-Jul. 2007
- [8] K. Müller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, T. Wiegand, "Multi-view Video Coding Based on H.264/AVC Using Hierarchical B-Frames," *Picture Coding Symposium*, 2006

Table 1. Comparison examples between view-first and time-first when different scenarios are utilized.

	(1)	(2)	(3)	(4)
view-first	44	117	44	32
time-first	44	56	56	23.5

Note: Scenarios through (1) to (4) are the same as those in Table 2. gl is 16 and nv is 8.