

## H.264 HIERARCHICAL P CODING IN THE CONTEXT OF ULTRA-LOW DELAY, LOW COMPLEXITY APPLICATIONS

*Danny Hong, Michael Horowitz, Alexandros Eleftheriadis, and Thomas Wiegand*

Vidyo, Inc.

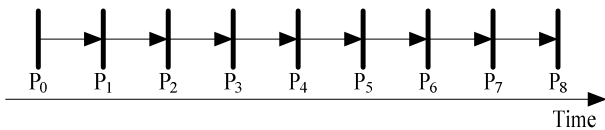
### ABSTRACT

Despite the attention that hierarchical B picture coding has received, little attention has been given to a related technique called hierarchical P picture coding. P picture only coding without reverse prediction is necessary in constrained bit rate applications that require ultra-low delay and/or low complexity, such as videoconferencing. Such systems, however, have been using the traditional IPPP picture coding structure almost exclusively. In this paper, we investigate the use of hierarchical P coding vs. traditional IPPP coding and demonstrate that it has significant advantages which have not yet been well documented or understood. From a pure coding efficiency point of view, we show that for encoders configured to use ultra-low delay and low complexity coding tools, hierarchical P coding achieves an average advantage of 7.86% BD-rate and 0.34 dB BD-SNR.

**Index Terms**— H.264, hierarchical P coding

### 1. INTRODUCTION

The temporal prediction structure for ultra-low delay (i.e., one-way delay less than 200 milliseconds) video coding has been traditionally the so-called IPPP coding structure as depicted in Fig. 1, with  $P_i$  marking a picture at time instant  $i$ . Further, the prediction structure is such that each P picture uses as a reference the picture that immediately precedes it, as shown in the figure. Each vertical line segment in the figure represents a picture and the arrows show the direction of prediction, from the reference picture to the input picture.

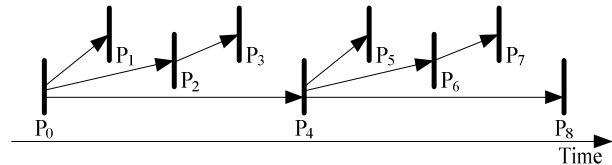


**Figure 1:** IPPP coding prediction structure.

It seemed almost set in stone that this was the only viable way of encoding video in low delay applications much as the IBBP structure was in broadcast encoding. In recent years, however, significant attention has been given to a temporal prediction structure called hierarchical B picture coding [1][2]. The bit rate reductions achievable

with hierarchical B picture vs. IBBP coding are typically on the order of 20%.

Interestingly, little attention has been given to a related technique for ultra-low delay video coding called hierarchical P picture coding. One example of a temporal prediction structure for hierarchical P picture coding is shown in Fig. 2. In the following, we will use ‘hPP’ to indicate a hierarchical P picture coding structure such as the one shown in the figure.



**Figure 2:** Hierarchical P coding prediction structure.

As some applications have rigid constraints on latency and/or computational complexity that prevent the use of B picture coding, including an ultra-low delay version of hierarchical B picture coding, it is particularly relevant to examine the relative benefits of hPP coding. Please note that any inter prediction reference that is in the future incurs a structural delay that is unacceptable for ultra-low delay applications and is therefore excluded here.

Many products in the videoconferencing industry today use H.264/AVC Baseline profile [3] together with IPPP. In this paper, we examine some advantages of hPP coding vs. traditional IPPP coding in the context of ultra-low delay, low complexity applications. The rest of this paper is organized as follows. In Section 2, an introduction to the concepts of hPP coding is presented. Section 3 contains a discussion of practical advantages that hPP coding has compared with IPPP coding. Some potential drawbacks are also discussed. In Section 4, results comparing hPP and IPPP coding efficiency in the context of an ultra-low delay and low complexity coding configuration are presented. We conclude with Section 5.

### 2. HIERARCHICAL P CODING

We begin this section with a review of traditional IPPP coding as depicted in Fig.1 above and leverage the nomenclature introduced to describe hPP coding. In the Baseline profile of H.264/AVC, two picture coding

techniques are available. The first, intra coding, predicts the content of the input video picture (i.e., the picture being coded) using decoded luma and chroma samples from regions in the picture that have been coded earlier (and would therefore be available to a decoder). The second, predictive coding, predicts the content of the input video picture using previously coded pictures called reference pictures. Note that the coding order and display orders need not be the same; in this paper, however, to ensure ultra-low delay operation, we only consider the case where the two are identical or, equivalently, the case where no reverse prediction is allowed.

A traditional videoconference will begin with an intra-coded or I picture followed by predicted or P pictures. Unlike applications requiring random access (e.g., digital video recordings) which insert periodic I pictures, a videoconference typically uses one I picture at the beginning of transmission and never again, except in special situations (e.g., error recovery).

hPP coding is similar to IPPP coding in all respects except for the prediction structure. In lieu of the “flat” prediction structure shown in Fig. 1, the prediction structure for hPP coding does not always reference the picture immediately preceding the input. Rather, the prediction structure is constructed so that the input pictures are decomposed into sets or temporal layers (TLs) that offer temporal scalability. Fig. 2 shows an example of an hPP prediction structure with a four-picture period. To emphasize the hierarchical nature of the structure, we have added a vertical offset to some of the pictures. In the figure, pictures  $P_0$ ,  $P_4$ , and  $P_8$  are in the lowest TL,  $P_2$  and  $P_6$  are in the middle TL, and the rest are in the highest TL. Note that prediction can only occur from a picture in the same or lower TL, and that the TLs have a regular structure in order to produce subsequences at different frame rates as described in more detail in Section 3.1.

### 3. HIERARCHICAL P VS. IPPP CODING

In this section, practical advantages of hPP coding over IPPP coding are discussed as well as potential drawbacks.

#### 3.1. Transrating without transcoding

For videoconferencing applications, probably the most important advantage that hPP coding offers over IPPP is the ability to adapt the frame rate, and consequently the bit rate, of a coded video sequence without having to perform cascaded transcoding. In cascaded transcoding, the original bitstream is decoded, pictures are discarded to achieve a desired frame rate, and the remaining pictures are re-encoded. Transcoding is needed when the IPPP picture coding structure is used, but is computationally expensive, introduces latency, and reduces coding efficiency [4][5].

Note that with hPP prediction structure such as the one shown in Fig. 2, the highest TL pictures are not used as

reference pictures and may be discarded, thereby halving the frame rate and reducing the bit rate of the coded sequence. Because they are not used as reference pictures, their removal does not affect the decoding of the remaining pictures. The same is not true with IPPP prediction structure. Likewise, once the highest TL pictures are discarded the next highest TL pictures are no longer used as reference and they too may be discarded, further halving the frame rate and reducing the bit rate of the coded sequence. Adapting frame and bit rate in this non-transcoding manner is computationally inexpensive, introduces negligible latency, and preserves video quality.

#### 3.2. Error resilience

In hPP coding, the fact that the TLs have unequal importance can be leveraged to produce enhanced error resilience. For example, if one were to use a forward error correction (FEC) technique (e.g., [6]) in the IPPP coding case, then every picture would need protection since loss of a single picture would result in error propagation to all subsequent pictures. Consequently, the bit rate overhead due to FEC would become a large fraction of the overall bit rate. In contrast, in the hPP coding case, one may choose to protect only the lowest TL pictures and conceal the loss of the upper TL pictures thereby reducing FEC overhead. Note that, as a result of the hPP structure, if none of the protected lowest TL pictures are lost there is no risk of error propagation since pictures in the lowest TL reference pictures from the same TL exclusively.

#### 3.3. Lower encoder complexity

An important and often overlooked advantage of hPP coding is the fact that the encoder does not need to reconstruct the highest TL pictures for use as motion compensated references. For example,  $P_1$ ,  $P_3$ ,  $P_5$ , and  $P_7$  in Fig. 2 do not need to be reconstructed. As a result, the computational resources used for the deblocking filter and, depending on implementation, sub-sample interpolation processes can be saved. Additionally, for samples that are not needed for intra prediction, inverse transform and de-quantization can be omitted. While this may not benefit an encoder implemented in hardware, it is advantageous for software-based encoders. Experimental measurements of execution times of two identical production-quality, ultra-low latency H.264/AVC encoders, one configured for hPP coding and the other configured for IPPP coding were collected for several sequences. The encoders were run on an Intel Core 2 Duo based laptop computer and each was configured to produce Baseline-conformant bitstreams. For each sequence, we ran the experiment at four different bit rates that closely corresponded to the QP values listed in [7]. The average encoding speed gain is listed in the last column of Table 1 where we see that overall, hPP coding yields a 9.31% speed gain over IPPP coding at the encoder.

### 3.4. Coding efficiency

Literature describing coding efficiency advantages of hPP coding versus IPPP coding is scarce. In [8], results are presented comparing hPP coding with a variety of QP scaling techniques (i.e., techniques in which different QP values are used for pictures in different TLs), all of which outperform IPPP coding. However, the work does not focus on ultra-low delay, low complexity applications. The work in [9] also demonstrates coding efficiency advantages of hPP coding over an ITU-T Video Coding Experts Group (VCEG) video sequence test set [7] containing a wide variety of source content, but little attention is paid to computational complexity. Some additional coding efficiency results for hPP coding are given in [10]. Section 4 of this paper contains experimental results comparing the coding efficiency of two identical H.264/AVC Baseline encoders except that one is configured to use the hPP prediction structure and the other the IPPP prediction structure. Both encoders are configured to use coding tools and parameter settings consistent with ultra-low delay and low complexity.

### 3.5. Potential drawbacks

Unlike IPPP coding, when using an hPP coding structure, pictures in different TLs have different importance and different temporal distances from their reference pictures. As a result, QP scaling is commonly used. Due to QP scaling, bits are unequally distributed among pictures and this has the potential to introduce latency in certain situations.

To understand how picture-to-picture fluctuations in coded picture size introduces delay, consider a constant bit rate (CBR) network link capable of transporting a coded video picture with size  $B$  bits in exactly one picture time interval, the time interval between video pictures in the original video sequence. In the case where all pictures have size  $B$ , the time required to transport the bits associated with each picture on the network is constant and equal to one picture interval. In the case where the pictures have unequal sizes but where the average picture size over the entire video sequence is  $B$ , some pictures will be larger than  $B$  and therefore take longer than one picture time to transport and others smaller and therefore take less than one picture time. In both cases, the total number of bits transported is the same, but the fact that the pictures become available and are displayed at regular time intervals (i.e., one picture time) implies that the system cannot take advantage of smaller coded pictures to reduce the delay incurred by the larger pictures. In practice, the delay due to QP scaling is typically a fraction of a picture time and represents a small fraction of the overall one-way delay. We emphasize that hPP coding with QP scaling has the potential to introduce latency only when the bit rate of the coded video is close to the capacity of the network link. Moreover, this behavior appears only

in fixed-rate, dedicated lines; Internet connections, in particular, rarely (if ever) exhibit these conditions.

A second, albeit minor, disadvantage of hPP is the fact that an encoder configured to use hPP requires additional memory for reasons described in detail in Section 4.1.1. hPP coding does not require additional memory at the decoder since decoders are required to allocate additional picture buffer memory as a matter of conformance.

## 4. CODING EFFICIENCY

In this section, we describe an experiment in which the coding efficiency of hPP and IPPP coding techniques are compared and present experimental results.

### 4.1. Experiment description

#### 4.1.1. Encoder software

The Joint Video Team (JVT) reference software JM 16.2 [11] was used to produce the results in this section. The software was modified to disallow the use of multiple reference pictures on a macroblock basis even when more than one reference picture is available. Note that the encoder requires a minimum of two reference pictures to perform the hPP coding shown in Fig. 2. For example, when encoding/decoding  $P_3$ ,  $P_2$  is used as a reference picture; however, it is necessary to save  $P_0$  for later  $P_4$  encoding. The modified code, encoder configurations and bit streams are available upon request [12].

#### 4.1.2. Encoder configuration

The common test conditions for the Baseline profile encoding specified in [7] were used, except for modifications to the following three parameters: *NumberReferenceFrames* is set to 2 instead of 4 (due to the code change described in Section 4.1.1, the encoder uses only one reference picture for motion search to ensure that hPP is not given an unfair advantage), *RDOptimization* is set to 0 for low complexity mode, and *UseRDOQuant* is set to 0 (as required when *RDOptimization* = 0).

The following parameters were added to the common conditions to support the hPP structure shown in Fig. 2.

- *LowDelay* = 1
- *PReplaceBSlice* = 1
- *HierarchicalCoding* = 3
- *ExplicitHierarchyFormat* = "p0e4t2p1r3t1p2e4t2"
- *ReferenceReorder* = 2
- *MemoryManagement* = 2

Given the above configuration, *NumberBFrames* = 3 is being configured for hPP coding and *NumberBFrames* = 0 for IPPP coding. Note that setting *PReplaceBSlice* = 1 ensures that only P pictures are used and the generated bit streams are H.264/AVC Baseline profile conformant. Additional details describing the above parameters may be found in the documentation included with the JM 16.2 software package [11].

For this experiment, we used the QP scaling specified in [9]. Let  $Q$  be the QP value used for the lowest TL P pictures (e.g., P<sub>4</sub>, P<sub>8</sub>, etc.). Then, the middle TL pictures are assigned QP value of  $Q+3$  and the highest TL pictures are assigned QP value of  $Q+4$ . The encoder configuration file is available upon request.

#### 4.1.3. Test sequences

The VCEG test sequences listed in [7] as well as four conversational application test sequences (Vidyo1, Vidyo2, Vidyo3, and Vidyo4) described in [13] were used. All video sequences are progressive scan in YCbCr, 4:2:0 format.

## 4.2. Experimental results

### 4.2.1. Objective results

Sequence	Format	FPS	Frames	BD-SNR	BD-Rate	Speed Gain
Container	QCIF	15	149	1.1	-21.7	11.05
Foreman	QCIF	15	149	-0.32	5.94	6.16
Silent Voice	QCIF	15	149	0.1	-1.98	10.33
Paris	CIF	15	149	0.15	-2.58	7.03
Foreman	CIF	30	297	0.14	-3.35	6.78
Mobile	CIF	30	297	1.26	-23.48	5.45
Tempete	CIF	30	257	0.73	-15.27	6.39
Big Ships	720p	60	149	0.85	-26.4	8.1
City	720p	60	149	0.82	-23.41	8.56
Crew	720p	60	149	-0.07	2.41	8.07
Night	720p	60	149	-0.17	4.53	7.74
Shuttle Start	720p	60	149	0.09	-2.48	10.1
Rolling Tomatoes	1080p	24	61	-0.1	4.91	11.02
Vidyo1	720p	60	597	0.3	-7.4	12.64
Vidyo2	720p	60	597	0.3	-7.98	12.5
Vidyo3	720p	60	597	0.27	-6.93	13.24
Vidyo4	720p	60	597	0.33	-8.5	13.19
<b>Average</b>				<b>0.34</b>	<b>-7.86</b>	<b>9.31</b>

**Table 1:** Coding gains (BD-SNR in dB and BD-rate in %) and speed gains (in %) of hPP relative to IPPP coding.

A summary of experimental results is presented in Table 1. Bjontegaard measurements [14] are used to compare coding performance between the two coding structures: positive BD-SNR and negative BD-rate values imply that hPP coding outperformed IPPP coding. We note that sequences that do not have scene changes or a large degree of motion (e.g., Container) benefit from hPP whereas sequences with more dynamic content (e.g., Foreman) do not. In practice, adaptive QP scaling could be used to mitigate losses of hPP in these cases.

### 4.2.2. Subjective results

The decoded video sequences associated with hPP coding were examined for evidence of undesirable visible artifacts (e.g., picture pulsing or beating) that may have been introduced by the hierarchical prediction structure with QP scaling. The results were consistent with those reported in [10] in that no undesirable artifacts were observed.

## 5. CONCLUSION

We have discussed the advantages of hPP coding as compared with IPPP coding in the context of ultra-low delay and low complexity applications. The advantages include the ability to transrate without transcoding, enhanced error resilience, reduced computational complexity for software implementations, and higher coding efficiency. In addition, we presented experimental results using JM 16.2 modified and configured to use coding tools typical of ultra-low delay, low complexity applications in which hPP coding yielded an average of 7.86% BD-rate and 0.34 dB BD-SNR advantage relative to IPPP coding. The coding gain is obtained without loss of subjective quality.

## 6. REFERENCES

- [1] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B Pictures," *Joint Video Team, JVT-P014*, Poland, July 2005.
- [2] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF," *IEEE ICME*, Toronto, Ontario, July 2006.
- [3] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for Generic Audio-Visual Services," *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, March 2009.
- [4] A. Eleftheriadis, R. Civanlar, and O. Shapiro, "Multipoint Videoconferencing with Scalable Video Coding," *Journal of Zhejiang University SCIENCE A*, Vol. 7, Nr. 5, April 2006, pp. 696-705. (Papers from the Packet Video 2006 Workshop.)
- [5] S-F. Chang and A. Eleftheriadis, "Error Accumulation of Repetitive Image Coding," *Proceedings, IEEE ISCAS*, London, England, May-June 1994, pp. 3.201-3.204.
- [6] A. Li, Ed., "RTP Payload Format for Generic Forward Error Correction," RFC 5109, December 2007.
- [7] T.K. Tan, G. Sullivan, and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 3", ITU-T SG16/Q.6, VCEG-AI10, 35<sup>th</sup> VCEG Meeting, Berlin, Germany, July 2008.
- [8] W. Wan, Y. Chen, Y.-K. Wang, M. M. Hannuksela, H. Li, and M. Gabbouj, "Efficient Hierarchical Inter Picture Coding for H.264/AVC Baseline Profile," *Proc. of Picture Coding Symposium*, May 2009.
- [9] M. Horowitz and D. Hong, "Use of Hierarchical P Picture Coding for EPVC Anchor 2", *ITU-T*, Question 6, Study Group 16, T09-SG16-C-0288, Geneva, Ch., October 2009.
- [10] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, 2007.
- [11] JM version 16.2 (<http://iphome.hhi.de/suehring/tml>).
- [12] Danny Hong (danny@vidyo.com).
- [13] "MPEG Verification Test for SVC" ([http://ip.hhi.de/imagecom\\_G1/savce/MPEG-Verification-Test/MPEG-Verification-Test.htm](http://ip.hhi.de/imagecom_G1/savce/MPEG-Verification-Test/MPEG-Verification-Test.htm)).
- [14] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves", *VCEG Contribution*, VCEG-M33, Austin, TX, April 2001.