SOURCE CHARACTERISTICS BASED FAST BITSTREAM SWITCHING

Bo Xie and Wenjun Zeng
PacketVideo Corporation
[xie, wzeng]@packetvideo.com

ABSTRACT

Bitstream switching is an effective way to deal with the bandwidth variation issue in transmitting multimedia over the Internet or wireless networks. This paper proposes a fast intelligent bitstream switching algorithm that avoids the problem of long delay and high complexity that often occurred in prior solutions. We show that it can achieve very close performance as an off-line MSE-optimized bitstream switching solution, with well-controlled quality drift. It is especially useful in the scenario of real-time multicasting over heterogeneous networks, where multiple bitstreams with different bitrates are generated on the fly and dynamic bitstream switching is required for individual clients.

1 INTRODUCTION

Bit rate scalability is essential when transmitting video over time-varying channels such as the Internet or the wireless networks. A bitstream that inherently has bit rate scalability is thus of importance. However, a scalable bitstream alone may not provide a large enough bit rate range to address large bandwidth variation without sacrificing the coding efficiency. In addition, current industrial standards, e.g., 3GPP, the third Generation Partnership Project, may not necessarily support scalable codecs. As a result, dynamic bitstream switching may be the only way to deal with bandwidth variation in a standard compliant way.

One way to do bitstream switching is to switch only at the I (Intra-coded) frames. This solution is simple, but either incurs long delay in the switching, or sacrifices the coding efficiency significantly, depending on how frequently I frames are encoded. Recently, S/SP/SI frames have been proposed to serve as special bridging frames for switching from one bitstream to another [1][2][3]. These specially designed bridging frames can reduce or completely eliminate the error drifting effect often observed in bitstream switching, but are typically very complex. They typically require generating two intermediate bridging streams between any two bitstreams, each requiring a re-encoding process. Therefore they may not be viable for real-time video communication. The storage overhead for the bridging streams is also of concern. In addition, the bit overhead for transmitting a S/SP/SI frame can be as large as transmitting an I-frame, which may have some negative impact in the bitstream switching scenario where the bandwidth is of the major concern.

The situation becomes much more complicated in the scenario of real-time multicasting over heterogeneous networks, where multiple bitstreams with different bit rates are generated on the fly, and bitstream switching is required at the server for each individual client. In this case, generating S/SP/SI frames on the fly for each individual client becomes prohibitively costly for the server. Restricting to switching only at I frames has the aforementioned delay and coding efficiency problem. Therefore, we propose to switch from one bitstream to another at potentially any time instance, without generating any intermediate bitstreams. One particular concern in such an approach is the introduced drifting effect due to the mismatch of the two reference frames associated with the switching in the two bitstreams. However, we feel that in the scenario of transmitting video over time-varying channels where packet loss is the major concern and quality variation is inevitable, a well-controlled quality drift due to bitstream switching is not a critical problem and is acceptable. After all, quality variation is inevitable due to large bandwidth variation. We propose a source characteristics based fast bitstream switching algorithm that can achieve well-controlled quality drift. In fact, in our subjective testing, well-controlled quality drift during bitstream streaming is often viewed as a nice graceful transition between two bitstreams with large quality difference, making the switching less noticeable, as opposed to the abrupt quality change when no drifting occurs.

2 SOURCE CHARACTERISTICS BASED FAST SWITCHING POINT SELECTION

We assume that the decision to switch the bitstream is made using some dynamic rate control algorithm based on estimating the instantaneous bandwidth. We also assume that once a decision to switch the bitstream is made, the actual switching action has to be done within a predefined switching window that controls the maximal delay allowed. The problem now can be formulated as performing switching point selection within the given switching window to reduce the quality drift as much as possible. The observation here is that well-controlled small quality drift is tolerable, at least perceptually. One way to solve this problem is to, for each time instance within the switching window, do actual switching at that time instance, and then decode/reconstruct the subsequent frames to measure the actual quality drift, e.g., in terms of mean square error (MSE). The time instance that results in the smallest quality drift is chosen as the switching point. This MSE optimized solution, however,
is very complex, not amenable to real-time application. Another extreme is to simply switch at the next frame whenever a switch decision is made, which apparently does not have any control on the drifting effect and may lead to unacceptable visual quality. Our goal is thus to come up with a lightweight switching point selection algorithm that reduces the quality drift as much as we can. We will, however, use the MSE optimized solution as the benchmark to evaluate our proposed solution.

2.1 Finding the switching point with a pair of most similar reference frames

The quality drift observed when switching to a non-I frame is a result of the reference frame change as shown in Fig.1. In general, two bitstreams, BS1 and BS2, could have the same or different frame rates, and any frame could be skipped during the encoding process. After the trigger point where a switch from BS1 to BS2 is requested, switching happens from the \((k+n)\)th frame in BS1 to the \((l+m+1)\)th frame in BS2, which is a non-I frame. Due to the bitstream switching, the reference frame of the \((l+m+1)\)th frame in BS2 changes from the \((l+m)\)th frame in BS2 to the \((k+n)\)th frame in BS1, as shown in Fig.1. Obviously, these two reference frames are different, which causes the quality loss in the reconstruction of the \((l+m+1)\)th frame in BS2. This quality loss will further propagate to the following frames until the next I frame is reached. This is referred to as quality drift. Therefore, to reduce the quality drift, a basic idea is to find a frame in BS2 within the switching window that has a pair of the most similar reference frames.

Specifically, suppose the trigger point is set after sending the \(k\)th frame in BS1 (see Fig.1), and the \(l\)th frame in BS2 is the most adjacent frame, i.e., \(TS^{(2)}(l-1) \leq TS^{(1)}(k) < TS^{(2)}(l)\), \(l > 1\), where \(TS^{(1)}(k)\), \(TS^{(2)}(l)\) and \(TS^{(2)}(l-1)\) are the timestamps of the \(k\)th frame in BS1, the \(l\)th and \((l-1)\)th frames in BS2 respectively. Suppose there are \(N\) frames in BS1 (including the \(k\)th frame) and \(M\) frames in BS2 (including the \((l-1)\)th frame) within the switching window. Let us define \(A_1\) and \(A_2\) to be the two sets of frame number in BS1 and BS2 within the switching window, i.e.,

\[ A_1 = \{k, k+1, \ldots, k+N-1\} \quad \text{and} \quad A_2 = \{l-1, l, \ldots, l+M-2\}. \]

The goal is to search the set \(A_1 \times A_2\) such that these two frames have identical or close timestamps whose difference is within a threshold. We define the threshold used in such a search as the smaller target frame interval of BS1 and BS2, i.e. search for \((i, j) \in A_1 \times A_2\), such that

\[ |TS^{(1)}(i) - TS^{(2)}(j)| \leq Th = \min \left( \frac{1}{FR^{(1)}}, \frac{1}{FR^{(2)}} \right). \]  

where \(FR^{(1)}\) and \(FR^{(2)}\) are the target frame rates of BS1 and BS2, respectively. The result of such a search is a set of candidate pairs of reference frames.

B. Coding complexity based similarity

After temporal similarity based search, there could be multiple candidates whose timestamps are identical or close. Next we make use of the source characteristics to refine the search. The basic idea is to evaluate the coding complexity of each candidate pair of reference frames, and find the reference frame pair with the most similar complexity among all the above candidates. The assumption here is that coding complexity is a good measure for the content similarity. The main goal is to differentiate a pair of frames that are close in both time and content from a pair of frames that are close in time but quite different in content (e.g., around scene change).

To that end, we take advantage of one property of the rate control algorithm we proposed previously [4] and that has been used to generate all the encoded bitstreams. In the rate control algorithm, we use MAD (Mean SAD, Sum of Absolute Difference) to quantify the coding complexity of the frames, which refers to the number of bits needed for encoding such frames to achieve constant quality. The frame level bit allocation is MAD based, and is defined as follow,

\[ T_n = C * \frac{\text{mad}_{n-1}}{\text{mad}_n} \]  

where \(C = \frac{BR}{FR}\), is the average target bits per frame; \(BR\) and \(FR\), respectively, are the target bit rate and frame rate of each bitstream; \(T_n\) is the target bit count for the current frame, \(\text{mad}_n\) is the MAD of the current frame, \(\text{mad}_{n-1}\) is the average MAD from the starting frame to the previous encoded frame. For details about the rate control algorithm, please refer to [4].
Based on Eq. (2), we have,
\[
\frac{T_n}{C} = \sqrt{\frac{\text{mad}_n}{\text{mad}_{n-1}}} \iff \frac{\text{mad}_n}{\text{mad}_{n-1}} = \left( \frac{T_n}{C} \right)^2 \tag{3}
\]
Since \( \frac{\text{mad}_n}{\text{mad}_{n-1}} \) shows the relative complexity of the \( n \)th frame with respect to all encoded frames, we use the similarity of \( \frac{\text{mad}_n}{\text{mad}_{n-1}} \) to evaluate the complexity/content similarity of two encoded frames in two different bitstreams. Eq. (3) shows that we can use \( \frac{T_n}{C} \) to represent the relative complexity equivalently. In the implementation, we use \( \frac{S_n}{C} \) (\( S_n \) is the actual frame size of the current frame) instead. Note that \( \frac{S_n}{C} \) should only be used for reference frame pairs with different timestamps to evaluate the content similarity. For reference frame pairs with identical timestamps, since they correspond to the same original frame, we will use the difference in \( S_n \) to evaluate the quality difference.

Specifically, the complexity similarity based search algorithm works as follows. Denote the set of frame sizes of the candidate pairs after temporal similarity based search as \( \{ (S_i^{(1)} , S_i^{(2)}) \}_{i=1}^{P} \) (see Fig.2), and \( A = \{ 1, 2, \ldots, P \} \), where \( S_i^{(1)} \) and \( S_i^{(2)} \) are the frame sizes of the \( i \)th candidate in BS1 and BS2, respectively. \( P \) is the total number of the candidates.

We search the candidate set \( \{ (S_i^{(1)} , S_i^{(2)}) \}_{i=1}^{P} \) to find
\[
i^* = \arg \min_{i \in A} \left| \frac{S_i^{(1)}}{C_1} - \frac{S_i^{(2)}}{C_2} \right|
\]
where the associated two reference frames have the most similar coding complexity, i.e., most similar content.

### 2.2 Implementation

In the implementation, we incorporate I-frame switching. Since I-frame switching does not cause any quality drift, it becomes our first choice in selecting the switching point. In addition, in the temporal similarity based search, we differentiate between the pairs of reference frames with identical timestamps, referred to as truly aligned reference pairs, and those with close timestamps, referred to as synchronization reference pairs. We will be biased towards the truly aligned reference pairs, i.e. if we find truly aligned reference pairs, we will do the refined search only on these aligned pairs.

Below is a summary of our implementation.

(i) Within the given switching window, search for I frames in the bitstream we are going to switch to. If there is an I frame, then stop the search and switch to that (first) I frame.

(ii) If there is no I frame within the switching window, then search for the truly aligned reference pairs. If there are some, then switch at the pair that has the smallest difference in the frame size \( S_n \).

(iii) If there is no truly aligned reference pairs, collect all the synchronization pairs based on Eq. (1) (with the threshold specified in Eq. (1), it is guaranteed to have some candidates), then find the best switching point based on Eq. (4).

### 3 EXPERIMENTAL RESULTS

We tested the proposed switching point selection algorithm and compared it with a simple switching scheme where the switching point is at the trigger point, and the MSE optimized switching scheme as described in Section 2. We tested on a number of MPEG4 sequences encoded at different bit rates and frame rates using the rate control algorithm we proposed in [4]. The I-frames are inserted at 10s intervals, i.e., at 0, 10s, 20s, etc time instance. Table 1 shows the switching table that specifies the timestamps for the trigger points (start_time). The bit rates and frame rates are reflected in the bitstream names. The switching window size is 1 second. It should be noted that within the switching windows, there is no I frames.

**Table 1: switching table for a talking head sequence:**

<table>
<thead>
<tr>
<th>Bitstream_name</th>
<th>start_time(ms)</th>
<th>end_time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_28kbps_3fps.mp4</td>
<td>0</td>
<td>15000</td>
</tr>
<tr>
<td>2_128kbps_15fps.mp4</td>
<td>15000</td>
<td>35000</td>
</tr>
<tr>
<td>1_28kbps_3fps.mp4</td>
<td>35000</td>
<td>45000</td>
</tr>
</tbody>
</table>

Figs. 3 and 5 show the comparison of the reconstructed frames using the simple switching scheme and our proposed algorithm. The simple switching scheme results in significant visual artifacts due to the mismatch of the reference frames. In this specific case, our proposed scheme can find the truly aligned reference frame pair. No annoying visual artifact is observed, even though the two reference frames have quite different quality. In fact, we have tested on many different sequences, and we have not observed any weird annoying visual artifact when the proposed switching scheme is used.

Figs. 4 and 6 show the corresponding PSNR curves. The proposed switching scheme provides nice gradual transition when switching from low bit rate stream to high bit rate stream, which makes the switching less noticeable. When switching from high bit rate stream to low bit rate stream, the transition is
more abrupt. Both cases show that the proposed switching scheme achieve close performance, in terms of quality drift, as the off-line MSE optimized switching scheme.

Fig. 7 shows another case where the “foreman” sequence was switched from the 20kbps @ 3fps bitstream to the 64kbps @10fps bitstream. The trigger point is at 2.4sec. In this case, there is NO truly aligned reference pair in the 1sec switching window. It is seen that the simple switching causes significant visual artifacts on the face due to the inevitable mismatch of the reference frames. Our proposed algorithm, however, makes use of the source characteristics to choose the most similar reference frame pair, resulting in no annoying visual artifact.

4 CONCLUSION

This paper proposes a fast intelligent source characteristics based bitstream switching algorithm that avoids the problem of long delay and high complexity typically observed in some prior solutions. We show that it provides much better video quality than a straightforward bitstream switching scheme, and close performance to the MSE optimized switching scheme. This suggests that without specially designed bridging frames, a decent solution that provides acceptable user experience is possible. This distortion-centric solution that aims to minimize the quality drift is only part of an on-going work that intends to provide a more complete solution that also takes into account the impact of the bandwidth overhead.

5 REFERENCES


Fig. 3: Reconstructed frames at the same timestamp after switching with the trigger point at 15000ms. Left: simple switching scheme; Right: proposed switching scheme.

Fig. 4: PSNR curve for three different switching schemes with trigger point at the 15s time instance.

Fig. 5: Reconstructed frames at the same timestamp after switching with the trigger point at 35000ms. Left: simple switching scheme; Right: proposed switching scheme.

Fig. 6: PSNR curve for three different switching schemes with trigger point at the 35s time instance.

Fig. 7: Reconstructed frames at the same timestamp after switching with the trigger point at 2400ms. Left: simple switching scheme; Right: proposed switching scheme.