variable frame skipping scheme based on estimated quality of non-coded frames at decoder for real-time block-based video coding

Tien-Ying Kuo, Yang Liang, and Chin-Cheng Chu

Department of Electrical Engineering, National Taipei University of Technology, Taiwan, R.O.C.
Email: tykuo@ee.ntut.edu.tw

Abstract

This paper proposes an encoder employing variable frame skipping (VFS) to improve the video quality in low bit rate channel. The basic idea of VFS mechanism is to decide and skip a suitable, non-fixed number of frames in temporal domain to reduce bit usage. The saved bits can be allocated to enhance the spatial quality of the video. In literature, several methods of frameskip decision have been proposed, but most of them only consider the similarities between neighboring coded frames as the decision criteria. Our proposed method takes into account the reconstruction of the skipped frames using motion-compensated frame interpolation at decoder. The proposed VFS models the reconstructed quality of the skipped frame and, therefore, can provide a fast estimate to the frameskip at encoder. The proposed VFS can decide the frameskip in real time, and its encoded video has better spatial-temporal bit allocation.

1. Introduction

Good video quality requires high channel bandwidth. However, channel bandwidth is often limited. Therefore, it is necessary to make the trade-off between video quality and channel bandwidth. Fixed frame skipping (FFS) is one of the approaches to meet the channel bandwidth. Although a larger fixed number of skipped frames can save more bits, unsmooth display occurs at high motion scene. Variable frame skipping can adjust the number of skipped frames depending on motion activity. Therefore, variable frame skipping can avoid unsmooth effect. In literature [1]-[3], there are several methods dealing with variable frame skipping. However, they are not sophisticated because they only consider the relation between frames at encoder.

Song [1] adopted histogram of difference image (HOD) as the base of distortion estimation for frameskip decision. Liu [2] proposed a trellis method considering the distortion of skipped frames and it is accurate for temporal-spatial bit rate control. Pejhan [3] used a mechanism for dynamically varying the frame rate to create a high quality bitstream as well as separate files containing motion vectors at lower frame rates. Liu [2] and Pejhan [3] both must encode/pre-analysis whole video sequence offline without real time. For real-time applications, Vetro [4] used a model to estimate the distortion of encoded frames and the distortion of skipped frames reconstructed by frame repetition (FR) at decoder. But the distortion model of skipped frames only suitable for whole video sequence with low motion. In whole video sequence with high motion or high/low mixed motion, the accuracy for the distortion model of skipped frames is not enough.

Thus, we propose a method of variable frame skipping which selects a suitable number of skipped frames according to the decoder video quality. The proposed method of variable frames skipping can achieve two goals. The first goal is to enhance average video quality. The second goal is to decrease the variation of video quality.

2. Proposed variable frame skip methods

We propose two methods, greedy variable frame skipping (GVFS), and fast variable frame skipping (FVFS). Both methods optimize frameskip based on the quality of skipped frames interpolated by a global motion-compensated frame interpolation, called GMCI [5], and the skipped frames can be reconstructed well by GMCI and played smoothly at decoder. The first method GVFS mainly bases on the peak signal noise ratio (PSNR) of skipped frames at decoder to optimize frameskip. GVFS can provide higher quality video and lower variance of video quality than the method of FFS. Furthermore, GVFS can serve as the upper limit among the performance of various VFS’s. In order to reduce the complexity of GVFS for real-time applications, we propose the second method FVFS. It estimates the PSNR of skipped frames by modeling GMCI to decide suitable frameskip.
2.1. Greedy variable frame skipping (GVFS)

2.1.1. Relation between frameskip and PSNR. We interpolate the skipped frames between current frame \( i \) and next encoded frame \((i+FS+1)\). We could find out that relation between \( FS \) and \( \Delta PSNR_{i} \) expressed as

\[
\Delta PSNR_{FS_{a}}(i) < \Delta PSNR_{FS_{b}}(i), \quad \text{for} \quad a < b,
\]

where \( \Delta PSNR_{FS}(i) \) is the distance between the PSNR of current frame \( i \) and the lowest PSNR of skipped frames interpolated by GMCI from frame \((i+1)\) to frame \((i+FS)\), and \( FS \) is the number of skipped frames. However, VFS can increase PSNR of encoded frames. Thus, the relation between \( PSNR_{FS}(i) \) and \( FS \) can be expressed as

\[
PSNR_{FS_{a}}(i) < PSNR_{FS_{b}}(i), \quad \text{for} \quad a < b,
\]

where \( PSNR_{FS}(i) \) is the PSNR of non-skipped frame \( i \) encoded by saved bits from skipped frames.

2.1.2. Procedure of GVFS. GVFS would decide optimal \( FS \) to encode current frame. This \( FS \) is decided depending on the quality of skipped frames interpolated by GMCI, and decoder adopted GMCI would produce best performance. The optimal \( FS \) decision of GVFS for current frame is depicted in Figure 1, and the details are described as follows.

Step 1: Initiate \( FS \) as one.

Step 2: Calculate the number of available bandwidth to encode current frame \( i \) and next frame \((i+FS+1)\).

The available bandwidth can be calculated by

\[
BW = \frac{\text{bitrate (bits/sec)}}{\text{framerate (frame/sec)}} \times (FS+1). \quad \text{(1)}
\]

Step 3: GMCI interpolates all the skipped frames from \((i+1)\) to \((i+FS)\).

Step 4: Find out \( \Delta PSNR_{FS}(i) \) according to PSNR of current frame \( i \) encoded by Step 2 and all the interpolated frames by Step 3.

Step 5: “Fs decide” would judge whether \( \Delta PSNR_{FS}(i) \) is acceptable by the following criteria:
- If \( \Delta PSNR_{FS}(i) > TH \), Subtract one from \( FS \) as our final decided frameskip Encode current frame \( i \), and start a new run.
- If \( \Delta PSNR_{FS}(i) < TH \), Add one to \( FS \) and return to Step 2. Since video quality could be classified according to levels of PSNR, the threshold \( TH \) may vary with levels of PSNR.

2.2. Fast variable frame skipping (FVFS)

2.2.1. Distortion model of GMCI. As indicated in Figure 2, distortion model is derived by the relation between motion vectors, where \( \varphi_{o} \) is the original current frame, \( \varphi_{n} \) is the original next frame, \( \hat{\varphi}_{o} \) is the reconstructed current frame, \( \hat{\varphi}_{n} \) is the reconstructed next frame, \( e \) is the difference between \( V_{o} \) and \( V_{n} \), \( V_{o} \) is the false motion vector from \( P_{2} \) to \( P_{1} \) at decoder, \( V_{n} \) is the true motion vector from \( P_{2} \) to \( P_{1} \) at encoder.

The original \( n \)th skipped frame \( \varphi_{o}(P) \) can be represented as

\[
\varphi_{o}(P) = \alpha \cdot \varphi_{o}(P + V_{n}) + \beta \cdot \varphi_{n}(P - V_{o}),
\]

And the estimated \( n \)th skipped frame \( \hat{\varphi}_{o}(P) \) can be represented as

\[
\hat{\varphi}_{o}(P) = \alpha \cdot \hat{\varphi}_{o}(P + V_{n}) + \beta \cdot \hat{\varphi}_{n}(P - V_{o}),
\]

Thus, the interpolation error of the \( n \)th skipped frame can be derived and calculated by

![Figure 1: The block diagram of GVFS](image)

![Figure 2: Motion trajectory for interpolation modeling](image)
\[ e(P) = \varphi_m(P) - \hat{\varphi}_m(P), \quad (2) \]
\[ \alpha = \frac{FS + 1 - n}{FS + 1}, \quad \beta = \frac{n}{FS + 1}. \]

\[ V_{r,j} = \beta \cdot (V + \varepsilon), \quad \text{and} \quad V_{r,k} = \alpha \cdot (V + \varepsilon), \]
where \( \alpha \) and \( \beta \) are weight values, \( V_{r,j} \) is the \( V \) from \( P \) to \( P_j \), \( V_{r,k} \) is the \( V \) from \( P_k \) to \( P \).

2.2.2. Procedure of FVFS. As shown in Figure 3, the selection of frameskip depends on the quality of skipped frames estimated by “distortion model” of GMCI, and decoder adopted GMCI would produce good performance in skipped frame. FVFS would select suitable \( FS \) to encode current frame \( i \) as the following steps.

**Figure 3:** The block diagram of FVFS

**Step 1:** Initiate \( FS = FS_j \), \( FS_j \) is the number of skipped frames adopted in the previous encoded frame \( j \).

**Step 2:** Calculate the number of available bandwidth in previous encode frame \( j \) via Equation (1) and use the bandwidth to encode current frame \( i \).

**Step 3:** Estimate the distortions of all skipped frames between previous encoded frame \( j \) and current frame \( i \) according to Equation (2).

**Step 4:** Find out \( \Delta PSNR_{FS} (j) \) according to PSNR of frame \( j \) by Step 2 and all estimated frames by Step 3.

**Step 5:** Decide and set a suitable \( FS \),

- If \( \Delta PSNR_{FS} (j) > TH \), Subtract one from \( FS \) because the quality of skipped frames exceeds our tolerance. Then, return to Step 3.
- If \( 0 < \Delta PSNR_{FS} (j) < TH \), the quality of skipped frames is in our tolerance and may be desirable. Therefore,
  - \( <i> \) when \( FS_j < FS \), increase \( FS \) by two.
  - \( <ii> \) otherwise, \( FS \) is selected for current frame, where \( FS_j \) is the number of skipped that adopted in the old previous encoded frame.

**Step 7:** Encode current frame \( i \) by the selected \( FS \), and start a new run.

3. Experimental results

Our experiment environment is based on H.263+ encoder of the Test Model Near-term (TMN8) in the ITU-LBC Experts Group.

3.1. Accuracy of distortion model

The result of testing the distortion of skipped frames that estimated by the proposed model is shown in Figure 4. We compare the actual distortion of the interpolated frames with their estimation. We notice that the estimated distortion can fairly track the actual distortion of skipped frames.

**Figure 4:** Comparisons between estimated and actual distortion of the second skipped frames in Foreman sequence at 64K bps.

3.2. Comparisons of video quality

At encoder, we use FFS (FFS0 ~ FFS5) , where FFS0 means the number of frameskip is fixed and equal to zero, and two proposed variable frame skipping systems, GVFS and FVFS. At decoder, we adopt FR or GMCI as our frame restoration mechanism. The result is shown in Figures 5. We discover that GVFS has the excellent average PSNR and low variance of video quality. We compare FFS2 with GVFS/FVFS at encoder, where GMCI is adopted at decoder. The results of GVFS/FVFS are shown in Figure 6. Since video quality changed suddenly in short time make annoyance of human vision [6], FFS2 is the worst and GVFS has the best performance. FVFS is the second but reach the similar performance as GVFS.
3.3. Comparisons of computational complexity

The comparison is based on the average encoding time per frame, including both coded and skipped frames. We run and average 100 tests to get the result shown in Table 1. Spent time of FVFS is similar to those of various FFSs', and is really shorter than that of GVFS. Therefore, the proposed method, FVFS, has better video quality and is very suitable for real-time applications.

4. Conclusion

Good video quality requires high quality while keeping the variation of quality minimized. Our proposed methods can suffice for this requirement in high and low motion sequences. GMCI modeling gives the advantage of low complexity to evaluate GMCI interpolated frame at encoder, therefore, FVFS is far faster than GVFS and suitable for real-time applications. The performance of FVFS can approach to that of GVFS because of the accuracy of model.

Table 1: In Pentium® 4 PC (2.4GHz), time (sec/frame) spent for each method is compared in 64Kbps channel bandwidth.

<table>
<thead>
<tr>
<th>Average Times (sec/frame)</th>
<th>FFS=0</th>
<th>FFS=7</th>
<th>GVFS</th>
<th>FVFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman (encoded frames)</td>
<td>0.125 (400)</td>
<td>0.017 (50)</td>
<td>0.493 (1451)</td>
<td>0.057 (473)</td>
</tr>
<tr>
<td>Suzie (encoded frames)</td>
<td>0.121 (150)</td>
<td>0.015 (18)</td>
<td>0.551 (647)</td>
<td>0.051 (141)</td>
</tr>
</tbody>
</table>

5. Acknowledgments

This work was supported by the Nation Science Council (NSC 92-2213-E-027-036).

6. References