MOTION-COMPENSATED FRAME INTERPOLATION SCHEME FOR H.263 CODEC

Tien-ying Kuo, JongWon Kim and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, California 90089-2564

ABSTRACT

The temporal jerkiness removal for the H.263/H.263+ video conferencing system is the main object of this research. We proposed a fast block-based motion-compensated frame interpolation (FMCI) scheme which does not require any motion search in the decoder end, and therefore the complexity is low enough for achieving the real-time frame rate increase. In order to further improve the visual quality, we proposed two add-on schemes to enhance the FMCI, the adaptive frame skip (AFS) scheme in the encoder and the hybrid scheme of frame interpolation (HFI) incorporating both frame repetition and FMCI in the decoder. By adopting the two add-on schemes, we can successfully avoid the failure of FMCI prediction and achieve the smooth playback.

1. INTRODUCTION

In this emerging information age, video conferencing has been gradually evolving from the room-based to desktop-based video conferencing (DVC) system due to the advance of cheaper hardware. The media communication channel used by the video conferencing system is also switched from the dedicated ISBN line to the low cost and handy channel of the plain old telephone (POTS) with or without connecting to the Internet. The POTS line provides a limited bandwidth of 28-50 kbps, which brings a new problem to the video conferencing system. Due to the limited bit rate budget provided by communication channels, a video codec has to sacrifice visual quality in order to cope with the low bit budget. Especially, for the quickly growing wireless mobile communication, the available bandwidth is even limited to 8 kbps. Generally speaking, this bandwidth is too low to provide good quality of service (QoS) even implemented with the state-of-the-art H.263/H.263+ video-conferencing standard.

Two approaches are often adopted to meet the limited bandwidth constraint via the sacrifice of the spatial and temporal quality, respectively. The degradation of spatial quality can be achieved by making the quantization step coarser, which however introduces blocking artifacts. These spatial artifacts can be reduced by using the post-filtering and the overlapped block motion compensation (OBMC) techniques. There are many papers in the literature dealing with the artifact removal problem. Another means is to downsample the temporal resolution (or, equivalently, lower the frame rate) to meet the constraint of the low bit budget. The second approach is more efficient than the first one since much more bits can be saved by downsampling the temporal resolution than doing this in the spatial domain. However, the low frame rate often causes motion jerkiness observed in the decoder. The solution of removing the jerkiness is to perform the video post-processing such as frame interpolation at the decoder.

In the current literature [1], [2] , [3] most algorithms of motion-compensated frame interpolation (MCI) are designed for the pixel-based or the mesh-based but not block-based motion-field. Besides, their complexity is too high to be used in real-time applications because they require the motion search to find out the true motion trajectory for each pixel or the nodal point during interpolation. Even though H.263/H.263+ provides the optional PB-frame mode to achieve a similar goal of frame interpolation, the PB-frame still requires extra bits to encode the B frame overhead and the optional B-frame motion vector. Besides, the PB-frame mode can only interpolate one B frame between two adjacent P frames. It cannot insert many frames as needed. Therefore, the development of a real-time block-based motion-compensated frame interpolation scheme is an interesting and important problem.

2. FAST BLOCK-BASED MOTION-COMPENSATED FRAME INTERPOLATION (FMCI)

The proposed FMCI is implemented in the decoder as a video post-processing unit, which is cascaded with the standard H.263/H.263+ decoder without changing the bitstream syntax. As shown in Figure 1, FMCI consists of three main units: motion-preprocessing, segmentation and MCI prediction. The motion-preprocessing unit is used to modify the block-based motion field to achieve a better frame interpolation result. Once the post-processed motion field is obtained, we map it to the pixel-based motion field for MCI prediction. We adopt the deformable block transform to map the block-based motion field to the pixel-based motion field. The second unit of FMCI performs object segmentation of decoded frames, which is useful in providing the moving object location to MCI. We do not use any complicated segmentation procedure, partly because we do not want to increase the computational load in the decoder and partly because the segmentation result is rough due to the use of the block-based motion field only. For the third unit, classification of regions into stationary SB, covered CB and uncovered backgrounds UB and the moving object MO, which is used in standard MCI [1], is adopted here. The MO is predicted by the bi-directional prediction along the motion trajectories, while the SB , CB and UB are predicted by bi-directional, forward and backward prediction along the same spatial locations.

It is important how to map the block-based motion field to the pixel-based motion field. If FMCI scheme does not deform the block, it is easy to map the motion fields, that is, we can simply assign the pixel motion vector using the block motion vector of the associated block. However, the blocking artifacts will be observed in some cases due to the translation of a rigid block. In order to remove this kind of blocking artifacts, we should consider the block rotational motion and the mapping from a rectangle to a parallelogram as well. A deformable mapping, the affine transform, is

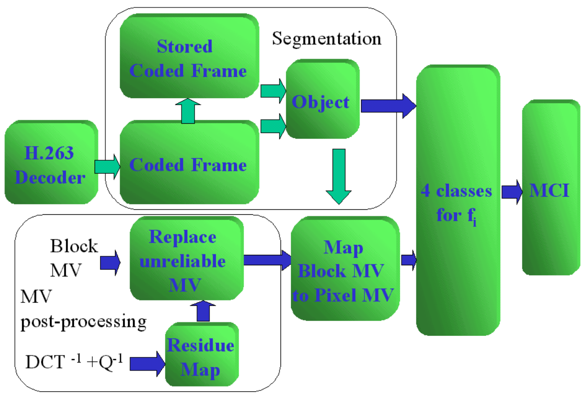


Figure 1: The block diagram of proposed FMCI implemented in the H.263/H.263+ decoder as a video post-processing unit.

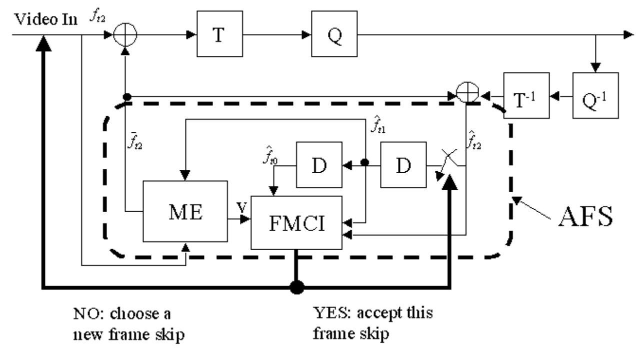


Figure 2: The block diagram of the proposed AFS scheme implemented as a feedback loop in the H.263/H.263+ encoder, where Q, T, ME and D denote quantization, transform coding, motion estimation, and time delay, respectively.

used in our mapping approach. The detail FMCI algorithm can be found in our previous work [5].

3. TWO ADD-ON FMCI ENHANCEMENT SCHEMES

We propose two add-on FMCI schemes to enhance the performance of FMCI interpolation. The first proposed scheme is for the encoder called the adaptive frame skip (AFS) scheme. We know that the proposed FMCI described in Section 2 implemented in the decoder and works independently of the encoder. That is, it can work with any standard H.263 bit stream without requiring a particular motion search or a frame skipping scheme in the encoder. Since the FMCI scheme can insert an arbitrary number of interpolated frames in the decoder, we can use an adaptive frame skip scheme (AFS), which adaptively chooses a variable number of skipped frame to result in a variable frame rate coding method. The constant frame rate playback can be recovered in the FMCI decoder. The second add-on scheme is called hybrid mode of frame interpolation (HFI), which interpolates frames by adaptively adopting either frame repetition (FR) or FMCI in the decoder. The purpose of HFI scheme is not to alter the standard encoder structure while avoiding the failure of FMCI prediction and preserving certain visual quality.

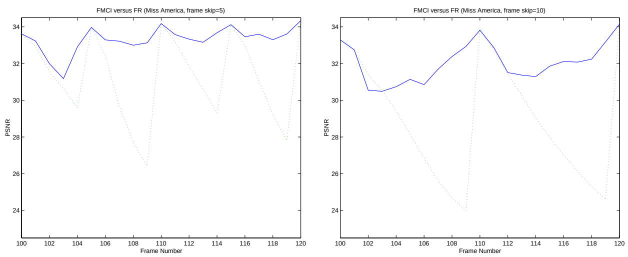
3.1. FMCI ENCODER WITH ADAPTIVE FRAME SKIP (AFS) SCHEME

To design the AFS scheme with a variable frame rate in the encoder, we should fully exploit the FMCI scheme adopted by the decoder. For some image sequences such as the Suzie sequence, the activity of the object movement varies from time to time. It can be fast in some frames but slow in others. If the entire sequence is encoded with a fixed frame skip, the power of the proposed FMCI scheme is not fully utilized. With the fixed frame skip, bits allocated to low activity frames are wasted, since low activity frames can be well predicted even with a larger frame skip. The fixed frame skip also causes poor results of FMCI prediction for high activity frames, where a smaller frame skip number should be used. Thus, we are motivated to develop a smart and adaptive frame skip (AFS) scheme in the encoder. Depending on whether the frame activity is low or high, AFS can adaptively adopt a larger or smaller value of frame skip. By the way, the saved bit budget can be reallocated to the coded image with a better coded image quality. Furthermore, better interpolated FMCI frames can be obtained due to the better quality of coded frames. The AFS scheme proposed in this section works seamlessly with the FMCI scheme of the decoder, and follows the standard of the H.263 encoder syntax and the bit stream syntax. That is, the frame skip value for each time coded frame is stored in the bit field called the Temporal Reference (TR) of the standard H.263 bit stream. Once we determine the TR value and generate the H.263 standard bit stream in the encoder, we can just apply FMCI to the received bit stream in the the decoder by inserting interpolated FMCI frames to restore the frame rate.

The implementation of AFS in the encoder is similar to that of the motion compensation unit which forms a feedback loop in the encoder as shown in Figure 2. We adopt the FMCI prediction inside the AFS unit to check whether the chosen frame skip value is suitable or not. As shown in Figure 2, if AFS determines that a frame skip number is not proper, it will continue to run the feedback loop until a suitable frame skip is decided.

One key difficulty in the design of AFS is that, since the jerkiness evaluation of FMCI is highly related to the human visual system, it is not trivial to find a criterion to evaluate the performance of FMCI inside AFS and then determine the frame skip. We know PSNR is popularly used as a metric in video or image processing, however, the goal of our frame interpolation is to allow the video object to move smoothly instead of precisely predicting the object location in non-coded frames of the original sequence. Therefore, PSNR is not a good metric to determine the visual performance of frame interpolation. However, PSNR can still be a tool to provide some useful information based on the following observation of testing and comparing PSNR curves of frame repetition (FR) and FMCI.

The PSNR performance of Frames 100 to 120 in the coding of the Miss America sequence is shown in Figs. 3(a) and (b) with the frame skip numbers of 5 and 10, respectively. We can see that in most cases the PSNR of FMCI is better than that of FR in both figures. Generally speaking, the PSNR curve of FR is a function of exponential decay while that of FMCI is a U-shaped curve. The PSNR curve of FR can be easily explained by considering the correlations of frames, in which the zero-order hold becomes weaker if the temporal difference becomes larger. For the



(a) (b)

Figure 3: Comparisons of the PSNR performance of FMCI (indicated by the solid line) and FR (indicated by the dotted line), where frames 100 to 120 of the Miss American sequence are tested with (a) a fixed frame skip of 5 (b) and a fixed frame skip of 10.

PSNR curve of FMCI, the U-shaped curve is a direct consequence from the use of bidirectional prediction of MCI. The bi-directional prediction will make the transition of interpolated frames much smoother. In addition, the weighting of two boundary coded pictures by the bi-directional prediction will pull up the saturation part of the PSNR curve of FR, because higher correlations between the current frame f_{t2} and the interpolated FMCI frame (e.g. f_{t2-1}) near the currently coded frame. However, in comparison with FR, the bi-directional prediction may pull down the PSNR performance of the interpolated FMCI frame (like f_{t1+1}) close to the previously coded frame a little more due to the low correlations between f_{t1+1} and f_{t2} and the non-true motion vectors used to traverse the motion trajectory. This is observed in frames 101-104 of Fig. 3(b). The difference between FR and FMCI in the first several interpolated frames is nevertheless smaller.

The above observation suggests that we can determine the proper frame skip inside the AFS feedback loop by measuring and comparing the PSNR performance of FR and FMCI on interpolated frames (like Frame 101-104/111-114) near the previous coded frame location. For example, as shown in Figures 3(a) and (b), the frame skip 10 for Frame 110 is too large because of the poor PSNR of FMCI in Frame 101-104, and then we should reduce the frame skip to 5 or the value between 5 to 10. In the following, we will describe an iterative algorithm to determine the frame skip in the AFS unit.

1. Select a smaller frame skip fs from a pre-defined frame skip set FS. We set $FS = 2, 4, 7, 10$. Only four elements are included in FS to avoid high computational complexity because each fs test requires a high-complexity motion search. We can choose one from the four possible values of frame skip in FS. Initially, We will choose the smallest fs in FS. Note that the largest element in FS is 10, which is large enough for any practical application (3 fps.) in the low bit rate coding.
2. Set the current frame $f_{t2} = f_{t1+fs}$. Use the reconstructed previous frame $\hat{f}_{t1}$ and the current frame f_{t2} to find the motion vector V_{t2} for f_{t2} and then obtain the reconstructed current frame $\hat{f}_{t2}$
3. Perform FMCI and FR based on $\hat{f}_{t1}$, $\hat{f}_{t2}$ and V_{t2}.
4. Determine whether fs is suitable or not. In the following three cases, we will choose fs as the final frame skip and terminate the iterations.
   - Compare PSNR of interpolated frames by using FMCI and FR. If the PSNR of FMCI is not smaller than that of FR by a threshold (i.e. fs for FMCI is good enough). We measure PSNR and the mean length of the motion field only in pixels belonging to the moving object MO. PSNR is measured on those frames belonging to the first one-third portions (f_{t1+1} to $f_{t1+\frac{t2-t1}{3}}$) of interpolated frames.
   - If the mean length of motion vector V_{t2} is larger than a pre-defined threshold (so as to avoid the loss of details of the object movement).
   - $fs = 10$, which is chosen to be the frame skip.

   Otherwise, we will go back to Step 1 and start a new iteration to choose a larger fs from FS.

3.2. THE HYBRID SCHEME OF FRAME INTERPOLATION (HFI)

The general encoder may not have ability to encode the bitstream with adaptive frame skip to optimize the FMCI performance. If we do not have permission to control the encoder, the encoded bitstream may contain the continuous frames with very different contents because large frame skip is adopted, the camera has big panning, zooming or encountering a abrupt scheme change. In this case, we may produce some poor visual artifacts in the decoder during the FMCI interpolation. To avoid this situation, we proposed a hybrid scheme of frame interpolation (HFI), which incorporates both the FMCI and FR in the decoder during interpolation. If the two continuous encoded frames has very different image contents, we adopts the FR interpolation, otherwise, this is the most cases, we will adopt the FMCI interpolation.

We may adopt the scene change detection technique to identify which HFI mode, FR or FMCI, should be chosen. Many scene change detection techniques is to detect the intensity change in the pixel domain or to check the occurrence of large motion field. However, those approaches are not suitable for the HFI, because those frames with the medium intensity change or motion vectors are where FMCI gains the most visual improvement. Instead, first we consider to traverse the motion trajectory inside the moving object and calculate the pixel intensity change in the boundary frames along the motion trajectory. If the averaged pixel intensity change along the motion trajectories is larger than a threshold, then the FR is applied, otherwise, we perform the next test in the histogram domain. We adopts the histogram approach [4] as our approach. We calculate the absolute differences of histogram bins $H_{y/cr/cb}(f_t, i)$ of current frame f_{t2} and previous frame f_{t1} in the Y, Cr and Cb domains ,respectively. We then add them together and normalized it by the histogram bin size 256 1. The normalized histogram sum HS is divided by previous normalized histogram sum to get a ratio HR. If either HS or HR is larger than the preset thresholds, then FI mode is used, otherwise, we will choose the FMCI mode.

$$\begin{aligned} HS_y(f_{t2}) &= \sum_{i=0}^{255} |H_y(f_{t1}, i) - H_y(f_{t2}, i)|, \qquad (1) \\ HS(f_{t2}) &= \frac{(HS_y + HS_{cr} + HS_{cb})^2}{256}, \\ HR(f_{t2}) &= \frac{HS(f_{t2})}{HS(f_{t1})} \end{aligned}$$

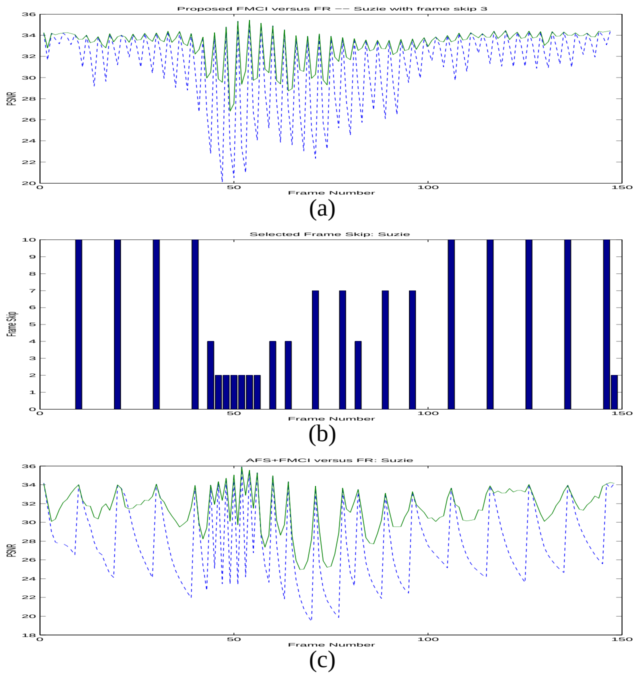


Figure 4: Performance of FMCI with/without AFS for Suzie sequence: (a) the PSNR performance of FR (indicated by dotted line) versus FMCI without AFS (indicated by solid line) using a fixed frame skip of 3 (b) selected frame skip of FMCI with AFS, and (c) the PSNR performance of FR (indicated dotted) versus FMCI with AFS (indicated by solid line) line) using variable frame skip as shown in (b)

4. EXPERIMENTAL RESULTS

Preliminary experiments were performed based on the TMN8 H.263+ video decoder with the replacement of frame repetition (FR) to the proposed FMCI scheme. We use the Suzie QCIF sequence as the test video to demonstrate the visual performance. In the encoder end, the original frame rate of the input sequence is 30 frame per second (fps), the basic mode (i.e. no optional mode is activated) is chosen, and the quantization step 20 and frame skip 3 are used. The required bandwidth for this encoded bit stream is about 23k bps. This bit stream will generate decoded video with 10 fps in the decoder. However, after inserting two interpolated FMCI frames, the frame rate can be restored to 30 fps, which is the same as the original video sequence.

skip, FR

The PSNR performance of FMCI without using AFS and HFI is shown in Fig. 4(a). We can see the FMCI can outperform the FR scheme. During actual video playback, it is easy to observe FMCI gives a better visual performance with more smooth motion and less artifact. In Fig. 4, we show the performance of the AFS scheme for the whole Suzie sequence, which adopts an adaptive frame skip with the fixed quantization level and without considering the rate-distortion constraint. Fig. 4(b) shows the frame skip selected by AFS as proposed in Section 3.1. The PSNR performance of the AFS scheme is shown in Fig. 4(c). We see that AFS can select the suitable frame skip to outperform FR in terms of PSNR. In the experiment, we found the HFI is useful in the large camera movement sequence, such as the foreman sequence. HFI will adopt FR in some periods, though in those periods, HFI cannot provide the smooth playback, it can avoid the severe artifacts caused by the poor FMCI prediction. The actual playback will be given in the conference.

It is worth mentioning that, in comparison with the fixed frame skip scheme, as shown in Fig 4(a), AFS chooses a larger frame skip (e.g. 10) to save the bit rate in frames no. 0-35 and 70-149. It also choose the smaller frame skip (e.g. 2) to avoid the poor FMCI prediction in the large motion part. Not that, even though we demonstrate the result of AFS by using the fixed quantization, AFS can be easily extended and seamlessly integrated with the rate-controller. For example, if AFS decides a larger frame skip than that decided by the rate-controller, we can just adopt the frame skip determined by AFS to save bits for enhance the quality. On the other hand, if AFS chooses a frame skip smaller than that determined by the rate-controller, we have two choices. The first one is to choose the frame skip decided by the rate-controller and then sacrifice temporal smoothness with the degraded FMCI interpolation. The second choice is to sacrifice the spatial resolution and to choose the AFS frame skip to achieve a better smooth playback. In this case, the coarser quantization will not degrade the visual effect a lot, since the smaller AFS frame skip usually occurs in the fast motion frames which will have the masking effect.

5. CONCLUSION

In this work, we developed a block-based fast motion compensated frame interpolation schemes for the video-conferencing system. FMCI uses the deformable block to perform interpolated moving object prediction. The interpolated frames by using FMCI efficiently increase the frame rate and remove motion jerkiness. AFS can successfully select the proper frame skip number by considering the FMCI performance. It attempts to avoid the prediction failure for a period with a very large motion and to increase the frame skip number for a period of slow motion to save bit rates. By adaptively adopting the FR and FMCI, the HFI can avoid poor visual effect due to the failure of FMCI interpolation prediction. The FMCI with AFS and HFI can provides a satisfying result in video playback.

6. REFERENCES

[1] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background", *Signal Processing: Image Compression 1*, pp.191-212, 1989.

[2] S. Tubaro and F. Rocca, "Motion estimators and their application to image interpolation", *Motion Analysis and Image Sequence Processing, Kluwer Academic Publishers*, 1993.

[3] T. Chen, Y. Wang, H. P. Graf, and Cassandra Swain, "A new frame interpolation scheme for talking head sequences", *Proc. of IEEE Intl. Conf. on Image Processing*, pp.591-594, 1995.

[4] J. Meng, Y.Juan and S.-F. Change, "Scene change detection in MPEG compressed video sequence", *IS&T/SPIE Symposium Proceedings*, vol.2419, Feb. 1995.

[5] Tien-ying Kuo and C.-C. J. Kuo, "Motion-Compensated Frame Interpolation for Low-Bit-Rate Video Quality Enhancement", *SPIE Electronic Imaging '99 - Visual Communications and Image Processing*, vol.3653, Jan. 1999.