FAST GLOBAL MOTION-COMPENSATED FRAME INTERPOLATOR FOR VERY LOW-BIT-RATE VIDEO QUALITY ENHANCEMENT

Tien-Ying Kuo and Lin-Ying Chuang

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

ABSTRACT

In this paper, we proposed a fast frame interpolator at decoder end to enhance the temporal quality of the real-time low-bit-rate video applications, such as the cellualr video conferencing. The proposed frame interpolator operates in the fashion of block-base instead of pixel-base to achieve the real-time speed. The global motion model is adopted in our scheme to interpolate the camera motion. Our scheme fully utilizes the received block-based motion vectors from the encoder to minimize the complexity of the global motion search. Furthermore, the majority of the local motion vectors for moving object interpolation can be directly derived without performing complex local motion search at the decoder end. The experiment shows that the proposed algorithm can interpolate frames in real-time and successfully remove the jigger artifacts caused by the low frame rate.

1. INTRODUCTION

A video coder usually sacrifices the visual quality to meet the bit budget constraint for very low bit rate applications.Even with H.263++ video coding standard, the visual quality of coded video is often unsatisfactory due to noticeable artifacts in the spatial domain and very low frame rates in the temporal domain. Spatial artifacts can be reduced by using the post-filtering techniques while the temporal resolution of the coded picture can be increased with a motion-compensated frame interpolation scheme. The primary objective of this work is to investigate the use of motion information contained in the H.263++ bitstream for quality enhancement of decoded video

The low frame rate often causes motion jerkiness observed in the decoder. One way to handle this problem is to increase the frame rate in the decoder to avoid jerky motion, which requires an effective frame interpolation scheme based on available transmitted (or decoded) frames. Three techniques have been widely used, namely, frame repetition (FR), frame averaging and motion-compensated frame interpolation (MCI)[1]. The frame repetition scheme duplicates the preceding decoded frame as the desired interpolating frame. It is the simplest method to increase the frame rate, but does not solve the problem of motion jerkiness. The frame averaging scheme interpolates frames using the averaged pixel intensity of the preceding and the succeeding decoded frames. It generally increases the PSNR value due to a better performance on the stationary part, but significant ghost artifacts are observed along the boundary regions of moving objects. Thus, it does not produce a good subjective result. It is well understood that the motion field information plays a critical role in the bit stream of low bit rate video. The technique of using the motion information to interpolate one or multiple frames between two consecutive decoded frames is called motion-compensated interpolation (MCI). MCI usually provides the best subjective result and is examined in this work.

This work was supported by National Science Council NSC 91-2219-E-027-001.

MCI was originally developed in the context of offline frame rate conversion, such as the conversion between different video or TV systems. A large amount of work has been done. Thoma and Bierling [1] proposed an MCI scheme by considering covered and uncovered backgrounds in addition to the stationary background and moving object. The original MCI interpolates frames by performing the motion search pixelwise for moving objects. The complexity is high. Many works [2][3] improved the original MCI by replacing it with block-base motion vector search to lower the operational complexity. However, those MCI variations still require motion search for interpolation and thus are not applicable to restore frame rate in the real-time video application, such as the video-conferencing. Our previous work [4] proposed a deformable MCI(DMCI) scheme to interpolate frames without performing motion search and, therefore, can restore frame rate in real-time. DMCI utilizes the block-based vector fields embedded in the bitstream from encoder and therefore the motion search is avoided. The DMCI warps the block with affine transform on the interpolated frame to avoid the blocky artifacts. However, some small perturbations on the received motion vectors due to camera motion may cause the slight “running water” artifacts and in the background and poor prediction in objects. Furthermore, DMCI requires three frames for moving object segmentation, which may cause the slight display latency. Migliorati et. al.[5] introduced the global motion estimation to the original MCI, however, in which all operations are in pixel-base and requires extremely high complexity for both the global and local motion search. In this paper, we retained the merits of low complexity given in DMCI scheme, while proposing an global motion-compensated frame interpolator (GMCI) by considering the global motion estimation to compensate the camera motion for enhancing the visual quality.

The paper is organized as follows. We first overview the concept of our proposed GMCI scheme and its framework in Section 2. The GMCI framework is further divided into two parts including the global motion estimation and the local motion-compensated frame interpolation, for which are presented in Sections 3 and 4, respectively. Experimental results and performance comparisons are provided in Section 5. Concluding remarks are given in Section 6.

2. SYSTEM OVERVIEW

The proposed GMCI scheme acts as a video post-processing unit at the block-based video decoder, which is able to use the conformed bitstream to interpolate frames without changing bitstream syntax. The major advantage of the GMCI system over other frame interpolators in the literature is to utilize the block-based motion fields $\mathbf{v_r}$ embedded in the bitstream so that the fast interpolation can be accomplished.

Figure 1 illustrates the proposed GMCI framework. The GMCI frame interpolator first decodes two frames from bitstream, called the previous frame F_{prev} and the current frame F_{curr}. The goal of the interpolator is to estimate frames F_{intp} in between to restore the frame rate. For providing more accurate estimation on F_{intp}, the interpolator has to estimate the global motion parameters f and $\mathbf{p}$ from both frames F_{prev} and F_{curr} by taking into account the camera's zooming and panning action. It is worthy of mentioning that the moving object O_{curr} can be identified as those regions of F_{curr} that do not follow the global motion model.

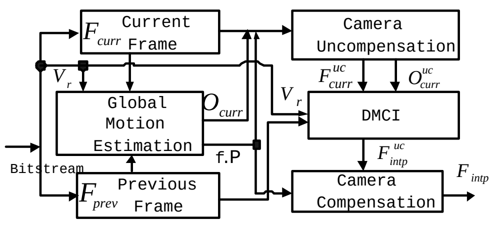

Fig. 1. Framework of GMCI Frame Interpolator

The system then remove the camera effects from the current frame F_{curr} and the segmentation map O_{curr} by un-compensating them with the estimated global motion parameters (f and $\mathbf{p}$), called the global un-compensated current frame $F^{uc}_{curr}$and the global un-compensated moving object O^{uc}_{curr}, respectively.

The next step is interpolate local motion-compensated frames F^{uc}_{intp} in between frames F_{prev} and F^{uc}_{curr}. The local motion-compensated frame interpolator performs DMCI[4] to interpolate frame, basically, by traversing local motion trajectory to warp objects. Then the local motion-compensated frames F^{uc}_{intp} can be obtained. In the last step, we compensate camera motion back to the local motion-compensated frames F^{uc}_{intp} to get the final GMCI interpolated frames F_{intp}.

3. GLOBAL MOTION ESTIMATION

Supposed that the whole picture content F_{curr} contains only the static background, that is, no moving object is observed from F_{prev} to F_{curr}, then any picture content inconsistence between F_{prev} and F_{curr} should result from camera zooming and panning action. The camera motion can be modeled [6] as

$$\mathbf{l^{uc}_{curr}} = f\,\mathbf{l_{curr}} + \mathbf{p} \tag{1}$$

where f is backward camera zooming factor, $\mathbf{p}$ is a two-dimensional backward camera panning vector, $\mathbf{l_{curr}}$ denotes a pixel location at F_{curr}, and $\mathbf{l^{uc}_{curr}}$ represents the corresponding location at F_{prev} before applying camera motion on $\mathbf{l_{curr}}$. Note that the origin of $\mathbf{l_{curr}}$ and $\mathbf{l_{curr}}$ locates at the center of the frame.

We perform global motion estimation to find the parameters f and $\mathbf{p}$ in Equation (1). Let $I_i(\mathbf{l})$ represent the intensity of pixel located at $\mathbf{l}$ of frame F_i. The global motion parameters can be obtained by minimizing the sum of intensity difference of $I_{prev}(\mathbf{l^{uc}_{curr}})$ and $I_{curr}(\mathbf{l_{curr}})$ for all frame pixels. Any kind of fast global motion search algorithms may apply to solve this problem. In this work, we adopt Tse's approach[6] to solve the global motion parameters,

$$\begin{aligned}
f^{(i)} &= \frac{N\sum < a^{(i-1)}, b^{(i-1)} > - < \sum a^{(i-1)}, \sum b^{(i-1)} >}{N\sum < b^{(i-1)}, b^{(i-1)} > - < \sum b^{(i-1)}, \sum b^{(i-1)} >} && (2)\\
\mathbf{p}^{(i)} &= \frac{1}{N}\left(\sum a^{(i-1)} - f^{(i)}\sum b^{(i-1)})\right), && (3)\\
a^{(i-1)} &= I_{prev}(\mathbf{l^{uc}_{curr}}) = f^{(i-1)}\,\mathbf{l_{curr}} + \mathbf{p}^{(i-1)},\\
b^{(i-1)} &= I_{curr}(\mathbf{l_{curr}})
\end{aligned}$$

where the summation performs over all frame pixels, N is the number of items summed inside each summation, the superscript (i) stands for the i-th iteration and $< \cdot, \cdot >$ denotes the inner product operation.

Since iterations are required for solving Equations (2)- (3), the complexity can be high. We reduce the complexity in two aspects. First, we do not sum all pixels, but only considering block central points as l_{curr} into the summation of Equations (2)- (3). For the QCIF format sequence(176 $\times$ 144 pixels) with macroblock size of 16 $\times$ 16 pixels, we reduce the sum of 176 $\times$ 144 items to 11 $\times$ 9 items. Secondly, since the block-based motion fields $\mathbf{v_r}$ from the encoder is available, we can initially iterate Equations (2)-(3) with the condition of $a^{(0)} = \mathbf{l^{uc}_{curr}} = \mathbf{l_{curr}} + \mathbf{v_r}$. The experiments shows that the iterations converge within two runs.

Since the actual picture content F_{curr} may contain moving objects in addition to the stationary background. Therefore, we have to exclude the blocks from the iterations if they falls into the category of moving object. Hence, at the end of each iteration, we have to look at Equation (1) by the iterated output of $f^{(i)}$ and $\mathbf{p}^{(i)}$. If the sum of the difference of $I_{prev}(\mathbf{l^{uc}_{curr}})$ and $I_{curr}(\mathbf{l_{curr}})$ is larger than a preset threshold, the block is classified to the moving objects, otherwise, the block belongs to the stationary background. Thus, in addition to the global motion parameters f and $\mathbf{p}$, the system produces the moving object segmentation map O_{curr} using only two decoded frames.

4. LOCAL MOTION-COMPENSATED FRAME INTERPOLATION

As shown in Figure 1, we have available outputs F_{prev}, F_{curr}, O_{curr}, f, $\mathbf{p}$ and $\mathbf{v_r}$ after performing the procedure described in Section 3. Before performing local motion-compensated frame interpolation, we first uncompensate and remove the camera motion on F_{curr} to obtain F^{uc}_{curr} using Equation (1). Then we use MCI technique in between frames F_{prev} and F^{uc}_{curr} to interpolate frames F^{uc}_{intp}, which is the interpolated frames without camera movement in action.

During MCI interpolation, we have to identify the object locations O^{uc}_{curr} on F^{uc}_{curr} by the help of O_{curr}. We divide the frame F^{uc}_{curr} into a macroblock grid, called control grid. For each grid block of F^{uc}_{curr}, we check its mapping position in F_{curr} using Equation (1). If one-quarter of mapping block covers O_{curr}, then we classify that block as O^{uc}_{curr}, or it falls into background categories.

After we categorize all blocks of F^{uc}_{curr} to either object O^{uc}_{curr} or background class, we furthure classify the background blocks into covered, uncovered and stationary one[1]. We adopt our previous work DMCI[4] approach here, which is a variation of MCI, to interpolate frames by averaging bi-directionally for stationary background, copying unilaterally for covered as well as uncovered backgrounds, and warping objects with affine transform using the control grid motion vectors for the objects. The control grid motion vector, or called local motion vector $\mathbf{v_l}$, describes the motion trajectory of the block-based object O^{uc}_{curr}. The control grid point is defined as one of four corners of the block-based object O^{uc}_{curr}. In order to find the grid motion vector $\mathbf{v_l}$ of each control grid point, we form an 8×8 block around the control grid point, and perform full motion search of it to find the best match on F_{prev}. However, we can derive the $\mathbf{v_l}$ using the available $\mathbf{v_r}$ to avoid the search complexity,

$$\begin{aligned} \mathbf{v_l} &= \mathbf{l_{prev}} - \mathbf{l^{uc}_{curr}}, \\ &= (\mathbf{v_r} + \mathbf{l_{curr}}) - (f\mathbf{l_{curr}} + \mathbf{p}), \\ &= (\mathbf{v_r} - \mathbf{p}) + (1-f)\mathbf{l_{curr}} \end{aligned} \quad (4)$$

The above derivation works under the condition that the camera zooming factor is not too big to deform block size too much, such that the $\mathbf{v_r}$ searched and provided by the encoder is reliable for deriving $\mathbf{v_l}$. We assume Equation (4) is valid only if the block is deformed within more one pixel by camera zooming, that is, $|BW - f \cdot BW| < 1$ pixel, where BW is the width of block and equals to 16. Thus, we say that Equation (4) is valid under the condition $0.937 < f < 1.063$.

Note that we have to find corresponding $\mathbf{v_r}$ of $\mathbf{v_l}$ to plug into Equations (4)-(5). Since the 8×8 block central at each grid point of F^{uc}_{curr} may map to and cover at most four block-based motion vectors $\mathbf{v_r}$'s of F_{curr}. We can perform the block matching test on those four candidates and pick the one giving the best match as our corresponding control grid vector $\mathbf{v_l}$.

Furthermore, since $(-176/2, -144/2) < \mathbf{l_{curr}} < (176/2, 144/2)$ for QCIF sequence, Equation (4) can be further simplified to

$$\mathbf{v_l} = \mathbf{v_r} - \mathbf{p} \quad (5)$$

if $(1-f) * 176/2 < \pm 1$ pixel, that is, $0.989 < f < 1.011$.

From Equations (4)-(5), we can simply derive the $\mathbf{v_l}$ via $\mathbf{v_r}$ with very low complexity except the camera has very heavy zooming ($f < 0.937$ or $f > 1.063$), for which we have to perform the full search for the control grid vector $\mathbf{v_l}$. Fortunately, our experiments show that, the probability for turning to full motion search is extremely low.

After we apply DMCI and interpolate the objects using warping to obtain the interpolated frame F^{uc}_{intp}. Then we should compensate the camera motion back onto F^{uc}_{intp} to obtain our target interpolated frame F_{intp}.

5. EXPERIMENTAL RESULTS

We test the MCI performance with the bit stream generated by the fixed frame skip scheme. Experiments are performed based on the UBC H.263++ video decoder with the replacement of frame repetition (FR) with the proposed GMCI scheme. We use the Foreman, Carphone, Miss America and Mother Daughter four QCIF sequences as the test video to demonstrate the visual performance. For each sequence in test, 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 2 are used in the encoder. The bitstream generates decoded video with 10 fps in the decoder. However, after inserting two interpolated GMCI frames, the frame rate can be restored to 30 fps, which is the same as the original input image sequence.

The averaged PSNR performance of GMCI for each test sequences is shown in Table 1. Three approaches are compared in the table. The averaged PSNR values show that GMCI outperforms both FR and DMCI schemes, especially, for the Foreman sequence in which obvious camera motion is observed. Figure 2 demonstrates PSNR curves for each frame in Foreman sequence. As we know that, in Foreman sequence, obvious camera motion is observed from frame 200-267. From Figure 2, we see GMCI in general provides improvement on PSNR over FR and DMCI between frames 200-267, while keeping the same PSNR performance as DMCI from frame 269-300.

Columns 2-4 in Table 2 show that the speed performance comparisons for three interpolators. GMCI only slightly increases the run time over FR and DMCI. The speed test is performed on the platform of the Pentiumn 4 2.0G Hz PC. During decoding playback on this platform, we can interpolate the GMCI frames in realtime. Columns 5-6 in Table 2 indicate the minimum and the maximum zooming factor f found in each sequence. Since all of the f's fall between 0.937 and 1.063, it indicates that the four test sequences all never turn to full motion search to obtain the local motion vectors, that is, all vectors can be calculated via Equation (4) or (5) with very low complexity.

We also provide the visual quality comparisons in Figure 3. Figure 3 demonstrates that GMCI is superior to DMCI, for example, in the area of the mouth and the upper left, upper center background. For the real-time playback, we see GMCI provide the smooth video with less "running water" artifacts in the background.

	FR	DMCI	GMCI
Foreman	26.60	28.62	28.74
Carphone	28.29	29.28	29.31
Miss America	34.72	35.84	35.85
Mother Daughter	30.92	31.53	31.53

unit: dB/frame

Table 1. Performance comparisons on averaged PSNR

	FR (sec /frame)	DMCI (sec /frame)	GMCI (sec /frame)	Min f	Max f
Foreman	0.033	0.042	0.043	0.996	1.006
Carphone	0.034	0.040	0.044	0.997	1.002
Miss America	0.030	0.038	0.039	0.999	1.002
Mother Daughter	0.034	0.040	0.041	0.996	1.003

Table 2. Performance comparisons on the speed test

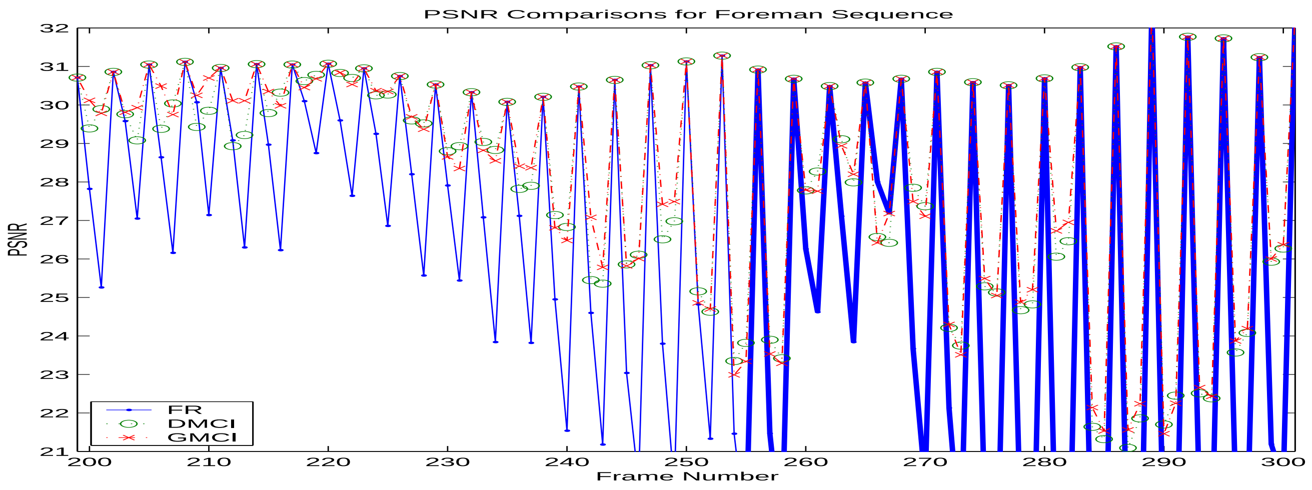

Fig. 2. PSNR comparisons for Foreman sequence (frame skip=2)

6. CONCLUSIONS

We developed a fast GMCI frame interpolator, which is able to restore frame rate at decoder in real-time. The low complexity is achieved partly because the interpolation operation is performed based on the block instead of pixel, and partly because we utilize the vector fields embedded in the bitstream to save the complexity on estimating both local and global motion parameters. The GMCI takes camera motion into account during interpolation. The experiments prove the proposed GMCI interpolator provide better performance over other interpolators.

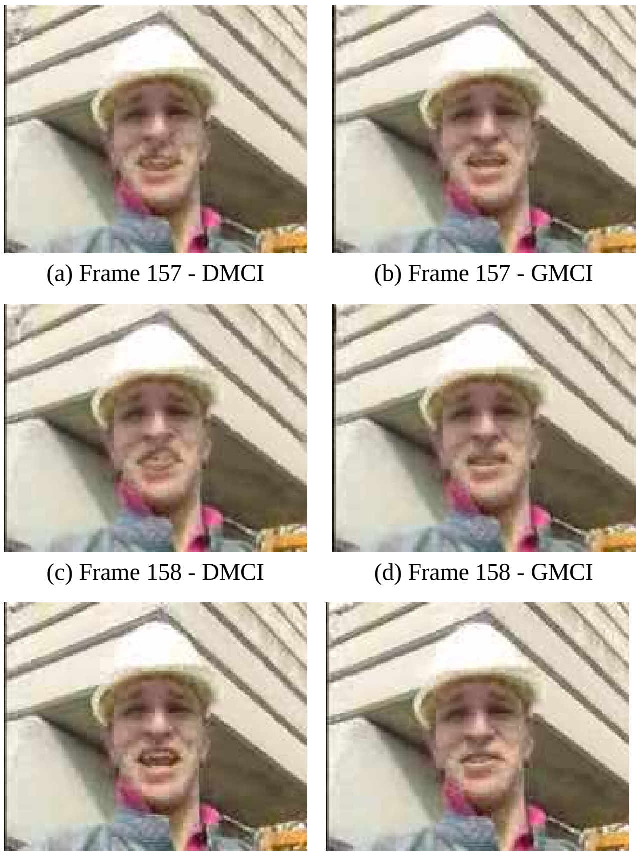

Fig. 3. Visual quality comparisons for consecutive interpolated frames (Frame 157-158) using decoded frames (Frame 156 and 159). The proposed GMCI successfully interpolates frames and provides better prediction in the area of mouth and upper left, upper center background.

7. REFERENCES

[1] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background," *Signal Processing: Image Compression 1*, , no. 191-212, 1989.

[2] C.K. Wong and Oscar C. Au, "Modified motion compensated temporal frame interpolation for very low bit rate video," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2329–2332, 1996.

[3] D. Bagni, V. Riva, and L. Albani, "An innovative temporal post-processing to increase the frame rate in h.263 encoding systems," *IEEE International Picture Coding Symposium*, pp. 273–280, April 1999.

[4] T.-Y. Kuo and C.-C. J. Kuo, "Improved h.263 video codec with motion-based frame interpolation," *Visual Communications and Image Processing*, vol. 3653, no. 1, pp. 26–39, January 1999.

[5] P. Migliorati, F. Pedersini, L. Sorcinelli, and S. Tubaro, "Semantic segmentation applied to image interpolation in the case of camera panning and zooming," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 25–28, 1993.

[6] Y. T. Tse and R. L. Baker, "Global zoom/pan estimation and compensation for video compression," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2725–2728, 1991.