2000年度 第13回 信號處理合同學術大會 論文集 第13卷 1號

양방향 움직임 추정을 이용한 효과적인 프레임 레이트 변환 알고리즘

최병태, 정유영, 고성제
고려대학교 전자공학과

Effective Frame Rate Up-conversion Using Bi-directional Motion Estimation

Byung-Tae Choi, You-Young Jung, and Sung-Jea Ko
Department of Electronics Engineering, Korea Univ.
E-mail : chois@dali.korea.ac.kr

Abstract— **We propose a new frame rate up-conversion algorithm for high quality video. In the proposed scheme, bi-directional motion estimation (ME) is performed to construct the motion vector (MV) field for the frame to be interpolated. Unlike conventional motion-compensated interpolation (MCI) algorithms, the proposed technique does not produce any overlapped pixel and hole region in the interpolated frame, and thus can utilize the overlapped block motion compensation technique to reduce the blocking artifacts. The proposed algoritm is very simple to implement on consumer products when compared to conventional MCI methods. Computer simulation shows a high visual performance of the proposed frame rate up-conversion algorithm.**

I. INTRODUCTION

Frame rate up-conversion is one of the main issues that have arisen in recent years with the explosive growth of image sources and display formats. For example, currently available motion pictures have a temporal rate of 24, 25, or 30 frames per second, while the HDTV and multimedia PC systems support higher temporal rates to reduce artifacts such as flicker and improve visual image quality [1]. Therefore, frame rate must be up-converted to use motion pictures in the HDTV or multimedia environments. Moreover, frame rate up-conversion technique can be used for video compression and slow motion replay.

Frame rate up-conversion algorithms such as frame repetition and linear interpolation by temporal filtering produce "jerkiness" into the motion portrayal and blurring of object boundaries, respectively [2]. It has been shown that the MCI technique provides the best solution in temporal rate up-sampling applications [3]. For the MCI scheme, the motion vectors must represent the true motion of objects in the image sequence since all the interpolation processes are controlled by the motion vectors. Several algorithms have been proposed for the true-motion estimation [4] - [6]. For example, Thoma and Bierling [4] use hierarchical block matching motion estimation to obtain motion vectors that more closely reflect the true motion of the displaced objects than the full search block-matching algorithm (FSBMA). Other techniques smooth the motion vector field in spatio-temporal directions to estimate the true object motion [6], [7].

Frame rate up-conversion using block-based motion compensation introduces the overlapped (multi-passing of motion trajectories) and hole (no motion trajectory is passing) regions in the interpolated frame. Kuo *et al.* used median filtering for the overlapped pixels [8]. To handle the holes, the spatial interpolation might be adopted [9]. This method, however, requires complicated operations since the spatial neighborhood of a hole may still contain other holes. Another method to fill the hole is to estimate the motion vector by using the neighboring motion field. A forward/backward prediction method based on the segmentation of the holes into the covered and uncovered regions has been developed under the assumption that the video sequence has static background [8], [10] - [11]. This method, however, produces unsatisfactory results especially in video sequences with camera motion such as panning and zooming which violate the assumption of the static background.

In this paper, a new frame rate up-conversion scheme is proposed to overcome the problem of the overlapped pixels and hole regions. In the proposed scheme, bi-directional motion estimation is performed using the existing previous and current frames to construct the motion vector field for the frame to be interpolated. Unlike conventional motion-compensated interpolation algorithms, the proposed technique does not produce any overlapped pixel and hole region in the interpolated frame, and thus can utilize the overlapped block motion compensation (OBMC) technique to reduce the blocking artifacts.

The paper is organized as follows. In the next section, a new frame rate rate up-conversion scheme is presented. An overlapped block MCI to reduce blocking artifacts is also presented in this section. Experimental results are given and discussed in section III. Finally, section IV concludes this paper.

II. PROPOSED FRAME RATE UP-CONVERSION USING BI-DIRECTIONAL MOTION ESTIMATION

Fig. 1 shows the block diagram of the overall motion-compensated frame rate up-conversion. The proposed scheme is composed of three processing units. First, bi-directional motion estimation block constructs the motion vector field in the to-be-interpolated frame. Second, the motion vector smoothing block smoothes the estimated motion vector in the spatio-temporal direction. Finally, overlapped block motion-compensated interpolation is performed to reduce blocking artifact of the block-based motion estimation and MCI. In the following subsections, we describe each functional block in detail.

A. Bi-directional motion estimation in the frame to be interpolated

Fig. 2 illustrates the proposed bi-directional motion estimation scheme. Each frame is first subsampled before initial motion estimation to reduce the computational complexity and to obtain smoothed motion vectors. For initial motion estimation,

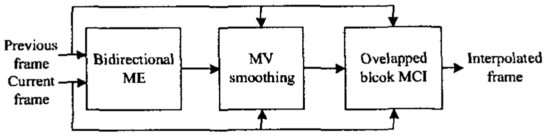


Fig. 1. Proposed frame rate up-conversion scheme.

the full search block-matching algorithm is used in the subsampled images. The estimated motion vector is used to initialize the initial value of the bi-directional motion vector without any modification.

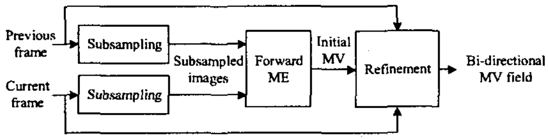


Fig. 2. Bi-directional motion estimation to construct the motion vector field in the to-be-interpolated frame.

In the next step, the initial motion vector is refined using the bi-directional motion estimation with a small search range in the full-scale image. Fig. 3 depicts the refinement process. First, consider the block B_i centered at $\vec{p}$ in the to-be-interpolated frame f_i between the frame f_1 and f_2. With the initial motion vector $\vec{D}_0(B_i)$, we search for the best linear motion trajectory passing through B_i using block-matching. The search range for B_1 in f_1 and B_2 in f_2 is confined to a small displacement $\pm d$ around the initial block position. The initial positions of the center pixels of blocks B_1 and B_2, respectively are calculated using the initial motion vector $\vec{D}_0(B_i)$ as follows

$$(x_1, y_1) = \vec{p} - \vec{D}_0(B_i), \qquad (1)$$

$$(x_2, y_2) = \vec{p} + \vec{D}_0(B_i), \qquad (2)$$

where $\vec{p} = (x_1, y_1)$ is the center position of the block B_i. Since the two compared blocks B_1 and B_2 moves simultaneously, the number of block matching becomes $(2d+1)^2 - 1$.

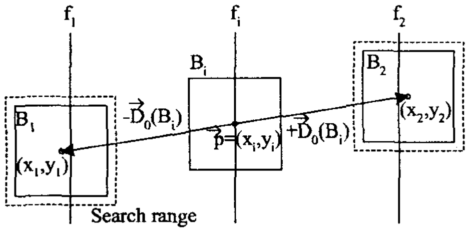


Fig. 3. Refinement of the initial motion vector using bi-directional motion estimation.

B. *Spatio-temporal smoothness on the bi-directional motion vector*

Once the bi-directional motion vector field is constructed, motion-compensated interpolation is performed to fill the intermediate frame. However, it is often observed that inconsistencies or non-smoothness in the estimated vector field decreases the interpolated picture quality severely. Inconsistencies of the motion vector can be corrected by constraining the spatio-temporal smoothness on the motion vector field.

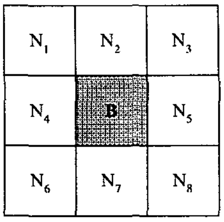


Fig. 4. Neighboring blocks used for motion vector refinement to guarantee the spatio-temporal smoothness.

Next we present the proposed smoothing scheme: Let B and N_i, where $i = 1, 2, ..., 8$ denote the current block and the eight nearest neighboring blocks around B in the to-be-interpolated frame as shown in Fig. 4, and $\vec{D}(B)$ and $\vec{D}(N_i)$ denote the corresponding bi-directional motion vector of B and N_i, respectively. Now, the DFD by the bi-directional motion vector $\vec{D}$ for the current block B can be defined by

$$DFD(\vec{D}, B) = \sum_{\vec{p} \in B} |f_1(\vec{p} - \vec{D}) - f_2(\vec{p} + \vec{D})|. \qquad (3)$$

Then, $\vec{D}_s(B)$, the spatio-temporally smoothed motion vector for B is obtained by

$$\vec{D}_s(B) = \arg\min_{\vec{D}} DFD(\vec{D}, B), \qquad (4)$$

where $\vec{D} \in \{\vec{D}(B), \vec{D}(N_i)\}$.

C. *Overlapped block MCI*

In order to construct the intermediate frame, interframe interpolation is performed using the motion vectors. In order to interpolate a block B in the to-be-interpolated frame f_i, straightforward block-based bi-directional motion-compensated averaging can be used as follows:

$$f_o(\vec{p}) = \frac{1}{2}[f_1(\vec{p} - \vec{D}(B)) + f_2(\vec{p} + \vec{D}(B))], \qquad (5)$$

where $\vec{p} \in B$ is the pixels in the to-be-interpolated block B. In the same manner, all pixel values in the frame f_i can be simply calculated on a block basis using (5). The main shortcoming of the above straightforward block-based MCI is the blocking artifact. It is often observed in the constructed frame when motion vectors are not correct or vectors in the neighborhood are significantly uncorrelated. In video coding, the OBMC technique is used to reduce the blocking artifact [12]. This technique, however, cannot be used in the conventional block-based MCI because the block grid in the interpolated frame is not contiguous due to the overlapped pixels and hole regions. Since the proposed frame rate up-conversion scheme produces the interpolated frame with non-overlapping contiguous block grid, the OBMC scheme can be incorporated with our MCI method.

The proposed MCI using the OBMC scheme employs a simple average interpolation method to reduce the computational complexity. For a given block B with $N \times N$ block size and a

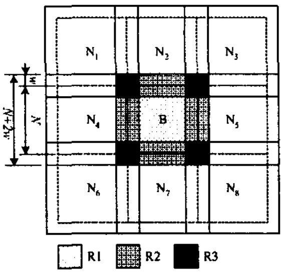


Fig. 5. Block overlapping pattern in the overlapped block MCI.

small overlapping width w, the original block size is extended to $(N+2w)\times(N+2w)$. Since the eight nearest neighboring blocks $N_i, i = 1, 2, ..., 8$, are also extended with the same size, three distinct overlapping regions $R1, R2$, and $R3$ as shown in Fig. 5 are generated. Let $f_o(\vec{p}, \vec{D})$ denote the motion-compensated averaging at $\vec{p}$ using the bi-directional motion vector $\vec{D}$. Then, the output of the overlapped block MCI in the extended block B is defined according to the number of block overlapping as follows:

1. For $R1$: No overlapping

$$f_o(\vec{p} \in R1, \vec{D}(B)) \tag{6}$$

2. For $R2$:

$$\frac{1}{2}\{f_o(\vec{p} \in R2, \vec{D}(B)) + f_o(\vec{p} \in R2, \vec{D}(N_i))\}, \tag{7}$$

where $N_i \in \{N_2, N_4, N_5, N_7\}$. 3. For $R3$:

$$\frac{1}{4}\{f_o(\vec{p} \in R3, \vec{D}(B)) + S_k\}, \text{for } k = 1, 2, 3, 4, \tag{8}$$

where S_k is the sum of the motion-compensated averaging for the neighboring blocks overlapped with B in R3 and defined by

$$\begin{aligned} S_1 &= f_o(\vec{p}, \vec{D}(N_1)) + f_o(\vec{p}, \vec{D}(N_2)) + f_o(\vec{p}, \vec{D}(N_4)), \\ S_2 &= f_o(\vec{p}, \vec{D}(N_2)) + f_o(\vec{p}, \vec{D}(N_3)) + f_o(\vec{p}, \vec{D}(N_5)), \\ S_3 &= f_o(\vec{p}, \vec{D}(N_4)) + f_o(\vec{p}, \vec{D}(N_6)) + f_o(\vec{p}, \vec{D}(N_7)), \\ S_4 &= f_o(\vec{p}, \vec{D}(N_5)) + f_o(\vec{p}, \vec{D}(N_7)) + f_o(\vec{p}, \vec{D}(N_8)). \end{aligned}$$

III. EXPERIMENTAL RESULTS

In this section, we illustrate some experimental results of the proposed algorithm. Four test sequences are used: `Suzie`, `Flower garden`, `Table tennis`, and `Beach`. Each sequence contains frames with a specific camera motion. Table I summarizes the test sequences and their characteristics.

TABLE I
FOUR TEST SEQUENCES.

Sequence	Frame size	Frame number	Typical camera motion
Suzie	176x144	150	No motion
Flower garden	352x240	115	Panning
Beach	352x240	184	Gradual scene change
Table tennis	352x240	150	Zooming and Abrupt scene change

The odd frames of these test sequences are eliminated for simulations. Conventional MCI algorithm consists of motion estimation using full search block matching, the refinement for spatial smoothness in [6], covered/uncovered classification in [11] for hole regions, and method in [13] for overlapped pixels. For each sequence, we perform 1:2 frame rate up-conversion and compute the average PSNR between the original and interpolated frames. The block size is 16 × 16 and the search range is ±16. The search range is from -2 to 2 in the motion estimation for the refinement of the initial motion vector. In overlapped block MCI, we use the overlapping width $w = 2$.

Fig. 6 shows the MCI results of the proposed algorithm for the `Table Tennis` sequence in zoom motion. Since the conventional method produces overlapped and hole regions as shown in Fig. 6(b), the interpolated image has annoying artifact along the edge of the table (see Fig. 6(c)). The proposed scheme shows good result in that area as shown in Fig 6(e). The values in the error image in Fig. 6(d) and (f) are multiplied by a factor of 4 for the better visual inspection.

Table II summarizes the simulation results of each test sequence. From the results, we can see that the performance enhancement of the sequences with camera motion is much more than that of `Suzie` sequence with no camera motion. Consequently, we can conclude that the proposed algorithm exhibits a better performance especially in sequence with camera motions. Note that, in the `Table tennis` sequence, abrupt scene change occurs two times. In that case, motion-compensated interpolation is switched off, and simple frame repetition is used.

TABLE II
PSNR COMPARISON.

Sequence	Conventional MCI	Proposed algorithm	Gain (dB)
Suzie	40.33	40.93	+0.60
Flower garden	27.22	29.07	+1.85
Beach	35.88	36.92	+1.04
Table tennis	29.38	30.84	+1.46

IV. CONCLUSIONS

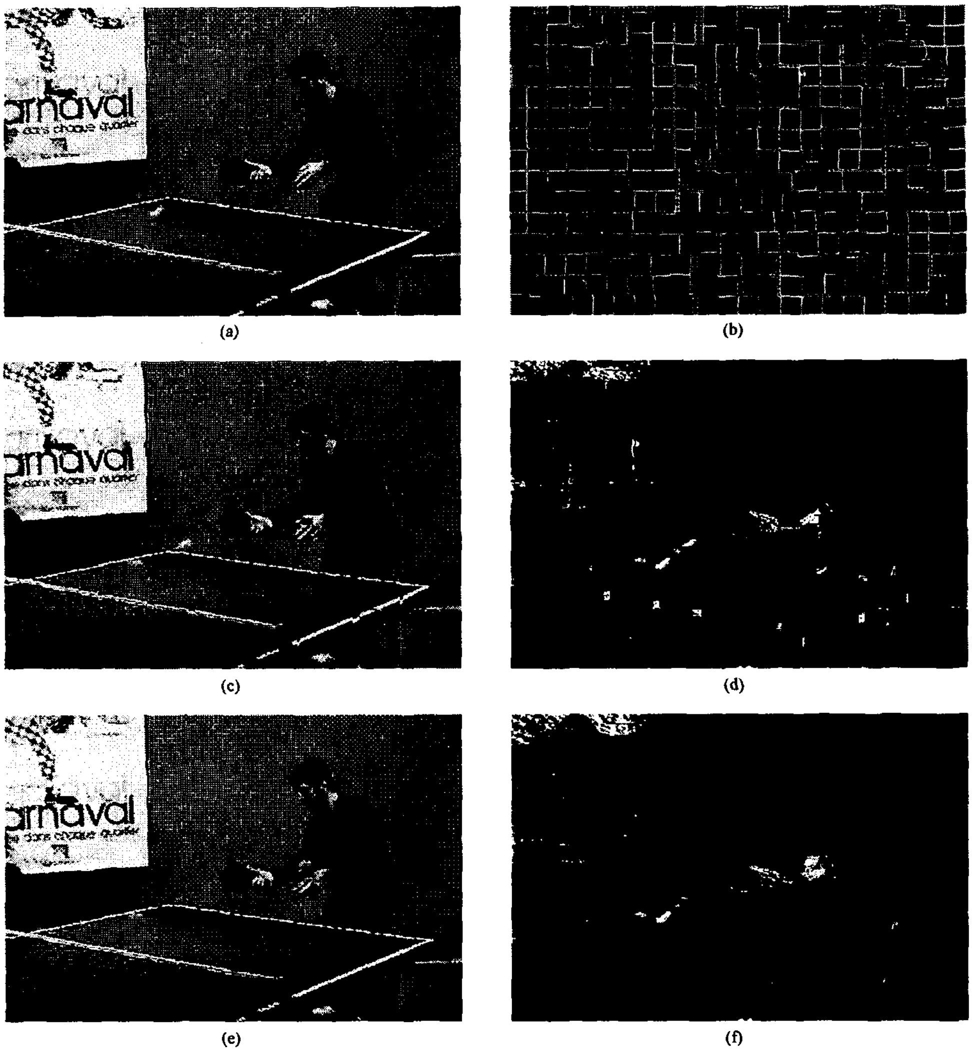
In this paper, a new motion-compensated interpolation algorithm for frame rate up-conversion using bi-directional motion estimation has been proposed. The proposed scheme is composed of three functional units; bi-directional motion estimation, spatio-temporal smoothing, and overlapped block motion compensated interpolation. The main feature of the proposed motion-compensated frame rate up-conversion scheme is that, unlike conventional MCI algorithms, the proposed technique does not produce any overlapped pixel and hole region in the interpolated frame. Moreover, by using the overlapped block motion compensation technique, the blocking artifact of the block-based motion estimation is effectively eliminated. It is very simple to implement the proposed MCI algorithm on consumer products when compared to conventional MCI methods due to its simplicity. Experimental results has proven that the proposed algorithm has better performance than the conventional MCI algorithm and is very robust especially in sequences with various camera motions like panning and zoom.

REFERENCES

[1] E. Petajan, "The HDTV Grand Alliance System," in *Proc. IEEE*, vol. 83, pp. 1094 1105, Jul. 1995.

[2] K. A. Bugwadia, E. D. Petajan, and N. N. Puri, "Progressive-Scan Rate Up-Conversion of 24/30 Source Materials for HDTV," *IEEE Trans. Consumer Electronics*, vol. 42, no. 3, pp. 312–321, Aug. 1996.

[3] S. C. Han and J. W. Woods, "Frame-rate Up-conversion Using Transmitted Motion and Segmentation Fields for Very Low Bit-rate Video Coding,"in *Proc. Int. Conf. Image Processing*, vol. 1, pp. 747–750, Oct. 1997.



(a) (b) (c) (d) (e) (f)

Fig. 6. Test sequence **Table Tennis**: (a) Original 81th frame, (b) overlapped and hole regions in the interpolated frame using the conventional MCI (c) interpolated image using the conventional MCI, (d) error image between (a) and (b), (e) interpolated image using the proposed algorithm, and (f) error image between (a) and (e).

[4] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background," *Signal Processing: Image Compression*, vol. 1, pp. 192 212, 1989.

[5] J. N. Youn, M. T. Sun, and C. W. Lin, "Motion estimation for high performance transcoding," *IEEE Trans. Consumer Electronics*, vol. 44, no. 3, pp. 649–658, 1998.

[6] G. D. Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[7] P. Moulin, R. Krishnamurthy, and J. W. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1606–1620, Dec. 1997.

[8] T. Y. Kuo and C. -C. J. Kuo, "Motion-compensated interpolation for low-bit-rate video quality enhancement," in *Proc. SPIE Visual Communications and Image Processing*, vol. 3460, pp. 277 288, July 1998.

[9] A. Kaup and T. Aach, "Efficient prediction of uncovered background in interframe coding using spatial extrapolation," in *Proc. ICASSP.*, vol. 5, pp. 501–504, 1994.

[10] R. J. Schutten and G. D. Haan, "Real-time 2-3 pull-down elimination applying motion estimation/compensation in a programmable device," *IEEE Trans. Consumer Electronics*, vol. 5, pp. 501–504, 1994.

[11] P. Csillag and L. Boroczky, "Enhancement of video data using motion-compensated postprocessing techniques,"in *Proc. ICASSP.*, vol. 4, pp. 2897 2900, 1997.

[12] T. Y. kuo and C. -C. J. Kuo, "Complexity reduction for overlapped block motion compensation (OBMC)," in *Proc. SPIE Visual Communication and Image Processing*, vol. 3024, pp. 303–314, Feb. 1997.

[13] D. W. Kim, J. T. Kim, and I. H. Ra, "A new video interpolation technique based on motion-adaptive subsampling," *IEEE Trans. Consumer Electronics*, vol. 45, no. 3, pp. 782 786, Aug. 1999.