A New Approach to Video Format Conversion Using Bidirectional Motion Estimation and Hybrid Error Concealment*

DANG-JIANG WANG AND JIN-JANG LEOU
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi, 621 Taiwan

To display different image/video sources on different display systems, video format conversion techniques are needed. In this study, a new approach to video format conversion using bidirectional motion estimation and hybrid error concealment is proposed. In the proposed approach, both the spatial information and the temporal information within an image sequence are employed. In particular, the temporal information obtained by means of bidirectional motion estimation is used in both interlaced-to-progressive conversion and frame rate conversion. A hybrid “error” concealment method developed by Chu and Leou [19] is modified to interpolate the remaining (unfilled) pixels within each intermediate frame. Based on the simulation results obtained in this study, the performance of the proposed approach is shown through comparison to be superior to (more robust than) that of several existing approaches.

Keywords: video format conversion, interlaced-to-progressive conversion, frame rate conversion, bidirectional motion estimation, hybrid error concealment

1. INTRODUCTION

To display different image/video resources on different display systems, video format conversion techniques are needed [1]. For an illustrated example between NTSC and HDTV, it is necessary to change their spatial resolution, scanning format, and frame rate. Video format conversion usually consists of three operations, namely, spatial resolution conversion, interlaced-to-progressive conversion, and frame rate conversion. Here the latter two types of video format conversion are treated.

For interlaced-to-progressive conversion (or deinterlacing), the existing methods can generally be classified into three categories: spatial deinterlacing techniques [2-5], temporal deinterlacing techniques, and hybrid deinterlacing techniques [6-11]. In spatial deinterlacing techniques, only the spatial information within the current field is used. The
two simplest spatial deinterlacing methods are line repetition and line average [2]. Doyle and Loynans [3] proposed an ELA (edge based line average) method to reconstruct missing lines, and their approach was modified by Kuo, Liao, and Lin [4]. Kim [5] proposed a deinterlacing algorithm based on sparse wide vector correlations. In temporal deinterlacing techniques, the two simplest temporal deinterlacing techniques are temporal repetition and temporal average [2]. In hybrid deinterlacing techniques, both the spatial and temporal information within the image sequence is used. The existing hybrid deinterlacing methods can generally be classified into two categories: filtering, and motion adaptive and motion compensation. Bock and Phillips [7] proposed a method which minimizes the artifacts caused by switching from one (moving) mode to another (stationary) mode while maintaining maximum vertical resolution in static and moving picture areas. Pohjala and Karlsson [8] presented a median based filter motion adaptive deinterlacing algorithm, whereas Lee, Park, and Kim [9] developed an improved motion-adaptive video format conversion method containing several modes. Huang and Chao [10] proposed a motion adaptive deinterlacing method, whereas Kovacevic, Safranek, and Yeh [11] presented an adaptive deinterlacing algorithm which successively builds approximations of the deinterlaced sequence by weighting various interpolation methods.


In this study, in interlaced-to-progressive conversion, bidirectional (forward and backward) motion estimation is performed. Bidirectional motion vectors are used to interpolate the missing lines in the input interlaced video signal. Each pixel in a missing line corresponds to one of the three possible cases: (1) one motion vector passes through the pixel, (2) multiple motion vectors pass through the pixel, and (3) no motion vector passes through the pixel. In the former two cases, the pixel can be interpolated by the two corresponding pixels in the previous and next fields forming the motion vector(s). In the third case, the pixel is linearly interpolated by the two corresponding pixels in the previous and next lines within the current field. In frame rate conversion, bidirectional motion vectors are similarly used to interpolate the intermediate (missing) frames. Then all the remaining (unfilled) pixels are interpolated using a modified hybrid “error” concealment approach proposed by Chu and Leou [19].
This paper is organized as follows. The proposed approach is presented in Section 2. Simulation results are given in Section 3, followed by concluding remarks.

2. PROPOSED APPROACH

2.1 Block-Based Bidirectional Motion Estimation

Block-based bidirectional motion estimation includes block-based forward and backward motion estimations. As shown in Fig. 1(a), to perform block-based forward motion estimation, for a pixel in the previous field, a matching block of size $7 \times 14$ centered at the pixel is formed. Within the matching block, only the original pixels in the previous interlaced field are included. The matching block within the previous field will search within a search window in the next field to find the "best" matching block with the minimum MAD (mean absolute distortion) value [1]. The search window is centered at the corresponding pixel within the next field of the central pixel of the matching block. Note that to determine the best matching block based on a given criterion, the MAD is perhaps the simplest and most commonly used one for best matching [1]. Backward motion estimation, as shown in Fig. 1(b), can be similarly performed. For a matching block, once its best-match block with the minimum MAD value is determined, a motion vector from the central pixel of the matching block in the previous (next) field to the central pixel of the best-match block in the next (previous) field is formed accordingly. Therefore, if the current field is an even field, then each pixel in the previous field (odd field) will have a motion vector from the previous field to the next filed, and vice versa.

2.2 Proposed Interlaced-to-Progressive Conversion Approach

In this study, as shown in Fig. 2, the proposed interlaced-to-progressive conversion approach consists of two processing steps described as follows.

![Fig. 1. Block-based bi-directional motion estimation.](attachment:image-url)
Fig. 1. (Cont’d) Block-based bi-directional motion estimation.

Fig. 2. The proposed approach to interlaced-to-progressive conversion.
Step 1. Perform block-based bidirectional motion estimation.

The procedure described in Section 2.1 is used to perform block-based bidirectional motion estimation.

Step 2. Check the condition of each pixel that is to be deinterlaced (interpolated).

Each pixel in the current field (frame) that is to be interpolated corresponds to one of the following three cases:

Case 1. Only one motion vector passes through the pixel to be interpolated. Then, if the corresponding MAD value for the motion vector is greater than a threshold, i.e., the best match is not good enough, then the pixel that is to be interpolated is re-classified as case 3 (described below); otherwise, the pixel that is to be interpolated is linearly interpolated by the two corresponding “matching” pixels in the previous and next fields, which form the motion vector.

Case 2. Multiple motion vectors pass through the pixel that is to be interpolated. Among the motion vectors, the motion vector with the minimal MAD value is selected. Then the pixel that is to be interpolated in the current field (frame) is processed using the same procedure employed in case 1.

Case 3. No motion vector passes through the pixel that is to be interpolated. In this case, the pixel that is to be interpolated is interpolated using a simple linear average technique, i.e., using the corresponding two pixels on the previous and next lines.

It is noted that (1) cases 1 and 2 will be re-classified as case 3 if the corresponding (minimal) MAD value is greater than a threshold; (2) if a motion vector passes through an existing pixel in the current field, then the existing pixel will not change; (3) the motion vectors employed in our three cases of motion vector conditions contain both the forward and backward vectors.

2.3 Proposed Frame Rate Conversion Approach

The proposed frame rate conversion approach is shown in Fig. 3. The motion vectors determined using bidirectional motion estimation are also used in interpolation of the intermediate (missing) frames in frame rate conversion. Each pixel within the intermediate (missing) frame can also be classified as corresponding to one of the three cases and interpolated using the techniques similar to that for cases 1 and 2 of the proposed interlaced-to-progressive conversion approach. Then the hybrid error concealment approach [19] is modified to interpolate (conceal) the remaining (unfilled) pixels.

Let \((x, y, t_n)\) be pixel \((x, y)\) within a frame \((t_n)\). To interpolate an intermediate (missing) frame using bidirectional motion estimation information, a pixel \((x, y, t_0)\) within a “stationary” or “homogeneous motion” area can be linearly interpolated using the two corresponding pixels, \((x, y-1, t_0)\) and \((x, y+1, t_0)\), in the previous and next lines, respectively, which have been interpolated. Denote the motion vector passing through pixel \((x, y-1, t_0)\) as \(V_1^{(V_1, V_{12})}\) and the motion vector passing through pixel \((x, y+1, t_0)\) as \(V_2^{(V_{12}, V_2)}\). If the two motion vectors, \(V_1\), \(V_2\), satisfy the following two conditions:
(1) \( |V_{m1} - V_{m2}| \leq 1 \) and (2) \( |V_{m1} - V_{m2}| \leq 1 \), then the pixel \((x, y, t_n)\) is treated as a “stationary” or “homogeneous motion” pixel, and is first linearly interpolated using the two corresponding pixels, \((x, y-1, t_n)\) and \((x, y+1, t_n)\). After all the “stationary” or “homogeneous motion” pixels are processed, the remaining (unfilled) pixels that are to be interpolated will finally be interpolated using a hybrid “error” concealment approach developed by Chu and Leou [19].

In this study, a block is considered to be 8×8 in size, and the believable neighboring blocks of a block represent all the completely interpolated blocks. A set of concealed block candidates for a partially-filled block, denoted as SC, is first generated, and then a proposed fitness function for frame rate conversion is used to select the “best” candidate among SC as the block candidate for the remaining (unfilled) pixels of the partially-filled block. Here the set of candidates, SC, for a partially-filled block is chosen from the previous and next frames within the corresponding extended area and the “average” block of the motion-compensated blocks (in the previous and next frames) of all the believable neighboring blocks of the partially-filled block.
In general, the boundaries between a block candidate and its believable 4-connected neighboring blocks are very smooth, a concealed block usually has statistical/spectral/spatial properties similar to those of its believable neighboring blocks, and a block usually has a motion vector similar to those of its believable neighboring blocks. Based on the above three observations, first the proposed fitness function CF will include the average inter-sample difference across the boundary (boundaries), AIDB, to evaluate the smoothness of the boundaries [19]. Second, to take the advantage of the statistical mean and variance, the proposed fitness function will also include two measures, namely, the average mean difference, AMD, and the average variance difference, AVD, between a block candidate and its believable neighboring blocks. Third, in terms of temporal correlation, a block candidate will be very similar to the motion-compensated block in the previous or next frame. The motion-compensated block S of a partially-filled block can be generated from the previous and next frames. Based on the experimental results obtained in this study, using either the “median” or the “average” of the motion vectors of the believable neighboring blocks of the partially-filled block will produce comparable results. Here the “average” motion vector is employed. Based on the motion-compensated block S, the difference SD(C, S) between a block candidate C and the motion-compensated block S of a partially-filled block was given in [19, 20]:

\[
SD(C, S) = \frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} |C(i, j) - S(i, j)|
\]

Combining the above four measures, AIDB, AMD, AVD, and SD, the fitness function CF for a block candidate C can be defined as follows:

\[
CF(C) = W_S \times SD(C, S) + W_B \times AIDB(C) + W_M \times AMD(C) + W_V \times AVD(C),
\]

where \(W_S, W_B, W_M\) and \(W_V\) are four weighting coefficients with \(W_S + W_B + W_M + W_V = 1\).

3. SIMULATION RESULTS

Four test image sequences, “Salesman,” “Claire,” “Miss America,” and “Table tennis,” were used to evaluate the performance of the proposed approach. The PSNR (peak signal to noise ratio) was employed in this study as the objective performance measure for the three components (Y, U, V) of an “interpolated” image. The PSNR of an interpolated image frame, denoted by PSNRp, is given by

\[
PSNR_p = \frac{4 \times PSNR_Y + PSNR_U + PSNR_V}{6},
\]

where \(PSNR_Y, PSNR_U, \) and \(PSNR_V\) are the corresponding PSNR values of the Y, U, and V components of an image frame, respectively.

To compare the performance of each of the existing interlaced-to-progressive (deinterlacing) approaches with that of the proposed approach, the four existing approaches were implemented in this study. They were: (1) the line repetition method, (2) the line average method, (3) the adaptive ELA algorithm [4], and (4) the motion adaptive dein-
Interlacing algorithm [10]. In the experiments, the interlaced version of the test images was created by simply dropping the even (or odd) lines of the original images. Based on the PSNR_p in dB, performance comparisons between the four existing interlaced-to-progressive conversion approaches and the proposed approach for the test image sequences, “Claire” and “Table Tennis” (the first 28 frames), are shown in Figs. 4 and 5, respectively. As a subjective measure of the quality of the deinterlaced images, the original image, the interlaced image, and the deinterlaced images treated using the four existing interlaced-to-progressive conversion approaches and the proposed approach for the 7th image (the Y component) frame of the image sequence, “Claire,” are shown in Fig. 6. Details of regions within the corresponding images illustrated in Fig. 6 are shown in Fig. 7. Similar simulation results for the other three image sequences can be found in [20], and thus are omitted here.

![Image](image1.png)

Fig. 4. The PSNR_p (dB) values for the first 28 frames in the “Claire” sequence of the four existing interlaced-to-progressive approaches and the proposed approach.

![Image](image2.png)

Fig. 5. The PSNR_p (dB) values for the 28 frames in the “Table Tennis” sequence of the four existing interlaced-to-progressive approaches and proposed approach.
Fig. 6. The original, interlaced, and deinterlaced images of the 7th image frame (the Y component) of the “Claire” sequence: (a) the original image, (b) the interlaced image, (c)-(g) the deinterlaced images using line repetition, line average, adaptive ELA, motion adaptive, and the proposed approach, respectively.
Fig. 6. (Cont’d) The original, interlaced, and deinterlaced images of the 7th image frame (the Y component) of the “Claire” sequence: (a) the original image, (b) the interlaced image, (c)-(g) the deinterlaced images using line repetition, line average, adaptive ELA, motion adaptive, and the proposed approach, respectively.

Fig. 7. Details of regions within the original, interlaced, and deinterlaced images of the 7th image frame (the Y component) of the “Claire” sequence illustrated in Fig. 6.
On the other hand, to compare the performance of existing frame rate conversion approaches and the proposed approach, the four existing approaches were implemented in this study. They were: (1) the field repetition method, (2) the frame repetition method, (3) the median filter based frame rate conversion method [12], and (4) the motion adaptive frame rate conversion algorithm [15]. In the experiments, the odd frames (frames 1, 3, …) were omitted from the original image sequence, and then the even frames (frames 0, 2, 4, …) were used to interpolate (reconstruct) the odd (missing) frames using the existing and proposed frame rate conversion approaches. Based on the PSNR_p in dB, performance comparisons between the four existing frame rate conversion approaches and the proposed approach for the test image sequences, “Claire” and “Table Tennis” (the first 14 interpolated frames), are shown in Figs. 8 and 9, respectively. As a subjective measure of the quality of the interpolated images, the original image and the interpolated images treated using the four existing frame rate conversion approaches and the proposed approach for the 5th image frame (the Y component) of the test image sequence “Claire” are shown in Fig. 10. Details of regions within the corresponding images illustrated in Fig. 10 are shown in Fig. 11. Similar simulation results for the other three image sequences can be found in [20], and thus are omitted here.

![PSNR_p (dB)](image)

Fig. 8. The PSNR_p (dB) values for the first 14 interpolated frames in the “Claire” sequence of four existing frame rate conversion approaches and proposed approach.
Fig. 9. The PSNR\(_p\) (dB) values for the 14 interpolated frames in the “Table Tennis” sequence of four existing frame rate conversion approaches and proposed approach.

Fig. 10. The original and interpolated images of the 5th image frame (the Y component) of the “Claire” sequence: (a) the original image, (b)-(f) the interpolated images using field repetition, frame repetition, median filter, motion adaptive, and the proposed approach, respectively.
4. CONCLUDING REMARKS

Based on the simulation results shown in Figs. 4–11 and the other simulation results obtained in this study [20], several observations can be made. (1) The corresponding results, based on the PSNRp (dB), for the proposed interlaced-to-progressive (deinterlacing) approach are better (more robust) than those for the four existing approaches, except for some particular frames. (2) The line average method is a good method for interpolating lines missing from interlaced images, but it blurs the edges of processed images. The adaptive ELA algorithm preserves the edges of processed images, but it may detect false edges, resulting in lower PSNR values for processed images. The motion adaptive deinterlacing algorithm is a good method for application to both moving and stationary pictures, but it requires the use of a precise motion detector. The proposed approach using bidirectional motion estimation can reduce false motions and obtain more precise motion vectors, resulting in better deinterlacing results. (3) The corresponding results, based on the PSNRp (dB), for the proposed frame rate conversion approach are better (more robust)
than those for the four existing approaches, except some particular frames. (4) The frame repetition method is a good method for application to stationary pictures, but it is not very suitable for moving pictures. The motion adaptive frame rate conversion algorithm is suitable for interpolating intermediate missing frames, but it may fail, due to the small search range for each motion trajectory. The performance of the proposed approach is superior to (more robust than) that of the four existing approaches.

Note that motion estimation takes the primary time complexity portions of both the proposed approach and the existing approaches using motion estimation information. For interlaced-to-progressive conversion, the time complexities of the line repetition method, the line average method, and the adaptive ELA algorithm [4] will be less than that of the motion adaptive deinterlacing algorithm [10] and the proposed approach, whereas the time complexity of the proposed approach is comparable to that of the motion adaptive deinterlacing algorithm. For frame rate conversion, the time complexities of the field repetition method, the frame repetition method, and the median filter based method [12] are less than that of the motion adaptive algorithm [15] and the proposed approach, whereas the time complexity of the proposed approach is comparable to that of the motion adaptive algorithm.

In summary, based on the simulation results obtained in this study, the performance of the proposed approach is superior to (more robust than) that of several existing approaches. This shows the feasibility of the proposed approach.

REFERENCES

10. C. L. Huang and T. T. Chao, “Motion compensated interpolation for scan rate

**Dang-Jiang Wang** (王淡江) was born in Taipei, Taiwan, Republic of China on Nov. 26, 1971. He received the B.S. degree in computer science and information engineering in 1995 from Feng Chia University, Taipei, Taiwan and the M.S. degree in computer science and information engineering in 1998 from National Chung Cheng University, Chiayi, Taiwan. His current research interests include image processing, pattern recognition, and computer vision.

**Jin-Jang Leou** (柳金章) was born in Chiayi, Taiwan, Republic of China on October 25, 1956. He received the B.S. degree in communication engineering in 1979, the M.S. degree in communication engineering in 1981, and the Ph.D. degree in electronics in 1989, all from National Chiao Tung University, Hsinchu, Taiwan.

From 1981 to 1983, he served in the Chinese Army as a Communication Officer. From 1983 to 1984, he was at National Chiao Tung University as a lecturer. Since August 1989, he has been on the faculty of the Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. His current research interests include image/video processing, visual communication, pattern recognition, and computer vision.