Novel Artifact Detection for Motion Compensated Deinterlacing

Arvin Chiang, and Yu-Cheng Fan, Member, IEEE
Integrated System Lab, Department of Electronic Engineering, National Taipei University of Technology
Taipei, Taiwan
arvin.chiang@hotmail.com and skystarfan@ntu.edu.tw

Abstract — Motion adaptive deinterlacing has been used widely in various kinds of commercial products, but the picture quality in motion areas still has much room for improvement. Motion compensated deinterlacing solves this problem by interpolation along the motion trajectory and thus is becoming the main trend of the next generation of deinterlacing algorithms. However, it suffers visible artifacts caused by incorrect motion vectors. To improve such imperfect pictures, this paper presents a motion compensation algorithm with more accurate motion estimation and more efficient artifact detection for digital television displays. Simulation results indicate that the proposed scheme can produce high-quality pictures with less flicker and imperceptible artifacts than a number of other techniques.

Index Terms— Artifact Detection, deinterlacing, display, motion compensation, progressive

I. INTRODUCTION

Various linear and non-linear methods have been proposed over the last decade to improve the quality of interpolation and trade it off with the implementation cost. Vertical spatial interpolation [1] exploits the correlation between vertically neighboring samples in a field when interpolating intermediate pixels. Defects occur with high vertical frequencies since vertical interpolation cannot discriminate between baseband and repeated spectra. Vertical temporal interpolation [1] is also a linear deinterlacing method that is designed such that the contribution from the neighboring fields is limited to the higher vertical frequencies. Nevertheless, degradation becomes visible at vertical edges. Edge-orientation dependent Interpolation [2] deinterlacing algorithms discussed so far is noted that interpolation along the direction of an edge can further improve the vertical resolution and reduce stair edges in the interpolated frame. Consequently, the picture quality of edge interpolation always relies on the correctness of edge detection. Motion adaptive deinterlacing [3] relies on accurate motion detection. Any erroneous detection will cause artifacts in the form of spots on the picture. Motion compensated deinterlacing [4] provides a solution to this problem by interpolating along the motion trajectory. The key goal of the motion compensated deinterlacing algorithm is to obtain a motion vector equivalent to the true motion and an error detection to estimate the correctness of the motion vector.

In order to solve the problems listed above, this paper presents a motion-compensated deinterlacing method using bidirectional interpolation, erroneous motion vector detection, flicker reduction, four-path temporal difference detection, weighting coefficient generation, cubic curve fitting interpolation, and fuzzy edge interpolation. This scheme proposes more accurate motion estimation and more efficient artifact detection so as to produce deinterlaced pictures with better visual quality, less flicker, and imperceptible artifacts.

II. PROPOSED MOTION-COMPENSATED DEINTERLACING

The block diagram of the proposed motion compensated deinterlacing method is shown in Fig.1. Bidirectional block-based motion estimation estimates bidirectional motion vectors of each block. Motion vector (MV) correctness detection determines whether the motion vector is close to the true motion or not. Weighting generation calculates the fading coefficient α for the soft switching between intra-field interpolation and inter-field interpolation. Temporal difference (TD) is calculated from frame difference and field difference from motion-compensation.

A. Improved Bidirectional Block-Based Motion Estimation

Block-based motion estimation is widely used in consumer multimedia products due to its relative computational simplicity. Its operation is described below.

1) Search Range and Block Size

In the case of deinterlacing, the out-of range block is interpolated by intra-field pixels, and ±32×16 pixels are required. High frequency components of a rapidly moving picture will lose their effect due to temporal accumulation, so a large search range is not required for compensation of resolution. Therefore, we define the search range as vertically 5 lines {+4, +2, 0, -2, -4} and horizontally 15 pixels {+7, …, 0, …, -7}. Based on the analysis in [1], temporal interpolation obtains the best results only when the vertical moving distance is just equal to even multiples of the spacing between two horizontal lines. Therefore, only even integers are considered in the vertical search range. Using smaller blocks is precluded in our design because of the insufficient number of pixels for the MV matching calculation and higher possibility of interference from background noise.

2) Bidirectional Interpolation
Bidirectional interpolation is the result of first order linear filtering according to the estimated motion vector. Its interpolated pixel \( P_i(x, y, n) \) can be formulated as:

\[
P_i(x, y, n) = \frac{P_0(x + d_x, y + d_y, n + 1) + P_0(x - d_x, y - d_y, n - 1)}{2}
\]

where motion vector is \( D = (d_x, d_y) \), as depicted in Fig. 2. Bidirectional interpolation requires a temporal delay of one field and provides two advantages. First, although the motion vector is not exactly equivalent to the true motion, bidirectional interpolation obtains smooth results, especially at the edge of an object. Second, frame difference can be calculated for judging the correctness of motion vectors. Therefore, only when and \( P_0(x + d_x, y + d_y, n + 1) \) and \( P_0(x - d_x, y - d_y, n - 1) \) is consistency, motion vector is judged to be correct. Both \( P_0(x + d_x, y + d_y, n + 1) \) and \( P_0(x - d_x, y - d_y, n - 1) \) were used to interpolated \( P_i(x, y, n) \).

![Fig. 2 Bi-directional motion vector.](image)

### B. Erroneous MV Detection and Flicker Reduction

MV correctness detection is achieved via four-path temporal difference detection with adaptive thresholds. In addition, an extra vertical high frequency detector is employed to further detect the errors that cannot be detected by the temporal difference method. Moreover, a local area expanded weighting coefficient generator is applied to tradeoff flicker and artifacts caused by erroneous MVs.

#### 1) Four-Path Temporal Difference Detector

A four-path temporal difference detector (4PTD Detector) is proposed to compensate for erroneous motion vectors and improve the reliability of detection results. As shown in Fig.3, the four paths are: six-point averaged SAD path, horizontally lowpass filtering (HLPF) path, horizontally highpass filtering and vertically highpass filtering (HHPF-VHPF) path, and horizontally highpass filtering and vertically lowpass filtering (HHPF-VLPF) path. The lower three paths in Fig.3 are constructed by an LPF (a 5-tap FIR with a zero placed at the subcarrier frequency), two subtractors, a 1H delay line, and an adder. The function of each threshold block is:

\[
\text{output} = 0, \quad \text{if input} \leq \text{threshold} \\
\text{output} = \text{input} - \text{threshold}, \quad \text{if input} > \text{threshold}
\]

![Fig. 3 Four-path temporal difference detection.](image)

Since each path extracts distinct characteristics of the local area, each threshold is designed differently to precisely correct the MV, and each threshold can be less critical in order to avoid detecting correct MVs as erroneous one. Moreover, temporal expansion is performed to deal with the “hole phenomenon” appearing on a fast moving object [2]. The judgment criterion for using temporal expansion is described as follows:

\[
\text{If MV of } P_i(x, y, n) = \text{MV of } P_0(x-d_x,y-d_y,n-1) \Rightarrow \text{ TD}(P_i(x, y, n)) = \max(\text{TD}(n-1,n+1), \text{TD}(n-2,n))
\]

else

\[
\text{TD}(P_i(x, y, n)) = \max(\text{TD}(n-1,n+1), \text{TD}(n-1(MC),n))
\]

where TD(n-1,n+1) is calculated from frame difference, TD(n-1(MC),n) is calculated from motion-compensated field difference, and TD(n-2,n) is calculated from frame difference of the last field.

#### 2) Adaptive Threshold of 4PTD Detector

For each pixel, SD (output of slope detector) and VE (output of vertical edge detector) are calculated first. Then, according to the value of SD and VE, the final threshold of each path can be determined by the block diagram in Fig.4. First, according to the motion vector, frame_FI(n,n-1) and frame_FI(n,n+1) are obtained by field insertion, as shown in Fig.5. In both frames, any high frequency slope transition of each interpolated pixel \( P_i(x, y, n) \), such as “+++” or “--”, can be detected by a slope detector, as shown in Fig.6. Its function is:

\[
\text{output} = -1, \quad \text{if input} < \text{threshold} \\
\text{output} = +1, \quad \text{if input} > \text{threshold}
\]

and the final SD of \( P_i(x, y, n) \) is calculated as:

\[
\text{SD}(P_i(x, y, n)) = \text{SD}(x,y,n,n-1) \oplus \text{SD}(x,y,n,n+1)
\]

where SD(x,y,n-1) and SD(x,y,n+1) are slope detector outputs of frame_FI(n,n-1) and frame_FI(n,n+1) respectively. If there is any slope transition of “+++” or “--” in frame_FI(n,n-1) or frame_FI(n,n+1), SD is equal to 1, and a lower threshold is selected. Otherwise, SD is equal to 0, and a higher threshold is selected. Since incorrect motion vectors often cause “+++” or “--” slope transition in frame_FI(n,n-1) or frame_FI(n,n+1), a lower threshold would contribute a higher TD to compensate for the incorrect motion vector. Although slope transitions of “+++” or “--” may also appear in detailed images of well interpolated pictures, the density of appearance is lower than that caused by erroneous interpolation. Therefore, the SD adaptive method is often used together with local area expanded weighting coefficient generation (see below) so that a better interpolation result can be obtained. Furthermore, the threshold selected by SD is scaled down by a factor VE calculated as follows:

\[
VE = \sum_{i=-1}^{1} \text{CL}(i) \times (P_i(x, y + 2i - 1, n) - P_i(x, y + 2i + 1, n))
\]

![Fig. 4 Adaptive threshold of 4PTD detector.](image)
undesirable. A vertical correlation detector, as shown in Fig.8, is used to detect this kind of artifact. Similar to slope detection, vertical correlation detection is processed on frame_FI(n,n-1) and frame_FI(n,n+1). Using frame_FI(n,n-1) as an example, if diff_1 and diff_2 are smaller, and diff_3 and diff_4 are larger than the threshold, then VC(x,y,n,n-1) is equal to 1, indicating that the vertical correlation of frame_FI(n,n-1) at P(x,y,n) is unreasonable. Otherwise VC(x,y,n,n-1) is equal to 0, indicating that the vertical correlation of frame_FI(n,n-1) at P(x,y,n) is reasonable. The function of the “threshold” block in Fig.8 is:

\[
\text{output} = \begin{cases} 
0 & \text{if input } \leq \text{threshold} \\
1 & \text{if input } > \text{threshold} 
\end{cases}
\]  

The final VC of P(x,y,n) is:

\[
VC(P(x,y,n)) = VC(n,n-1) \cdot VC(n,n+1)
\]  

III. SIMULATION RESULTS AND ANALYSES

In this paper, both MSE measurement and subjective view evaluation are used. As shown in Fig.9, MC deinterlacing is applied to re-conversion, and the re-converted pictures are compared with the 352×288 originally progressive ones through the quantities of MSE and PSNR. Subject picture qualities of MC-deinterlaced 352×288 originally interlaced pictures, such as resolution, alias, and block effect, are observed and evaluated.

Quantitatively, the PSNRs of our deinterlacing scheme for ten CIF sequences are compared with those of several other methods, as shown in Fig.10. Obviously, our method exhibits better results of PSNR performance than the other methods, even by 10dB in an extreme case.
Fig. 11(a) shows a weather sequence that has been deinterlaced by vertical interpolation. Fig. 11(b) shows the same parts of the same picture that has been deinterlaced by the proposed intra-field interpolation. It can be seen that the proposed intra-field interpolation provides smoother dominant edges and introduces less error at thin edges in Fig. 11(b).

The 21st reconstructed frames of the Stefan sequence are shown in Figure 12 (a)(b). The feathering effect caused by the fast moving object can be effectively detected by the proposed MV correctness detection algorithm. On the contrary, there are still some mis-detections in the picture de-interlaced in Fig. 12(a) by [6]. In addition, there are some defects in the white line due to the inaccurate motion vector. As a result, the pure inter-field interpolation causes large amounts of interpolation error in this situation. However, the proposed algorithm can accurately detect the unreliable motion information to prevent the possible resolution degradation in these critical frames.

The 289th reconstructed frames of the Mobile sequence are shown in Figure 12(c)(d). There are some artifacts in the Figure 12(c) since the rotational movement is not easily tracked by block based ME. The robustness of the proposed algorithm can be seen in Figure 12(d).

The computational complexity analysis will focus on the ME algorithm. For each block, the total number of operations required to estimate the MV is given in TABLE I, where $B$ is the number of pixels contained in the ME block and $K$ is the total number of candidate blocks in the search area.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Add</th>
<th>Sub</th>
<th>Mul</th>
<th>Shift</th>
<th>Abs</th>
<th>Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>≥3KB</td>
<td>3KB</td>
<td>-</td>
<td>-</td>
<td>3KB</td>
<td>-</td>
</tr>
</tbody>
</table>

* $B$=128 and $K$=153 in this work

**IV. CONCLUSION**

In this paper, we propose a high-quality deinterlacing algorithm using motion compensated interpolation. First, the bidirectional motion vector with high accuracy is estimated by interleaved sub-sample pseudo-frame SAD motion estimation. Second, the MV smoothing method is applied to ensure consistency in the MV fields. Third, MV correctness detection is used to prevent the artifacts caused by incorrect motion vectors. Finally, the coefficient for soft-switching is generated by a local area expanded weighting generator to tradeoff flicker and visible artifacts. Additionally, the spatial interpolation is further improved by the cubic curve fitting interpolation and the Fuzzy edge interpolation with error protection. As a result, the simulated PSNRs become 3-10dB better than those of previous deinterlacing methods. Also, it is verified that the proposed algorithm can provide higher picture quality and lower flicker for various kinds of video sequences.

**V. REFERENCES**


