A Threshold-Based Deinterlacing Algorithm Using Motion Compensation and Directional Interpolation

Hossein Mahvash Mohammadi[1], J.M. Pierre Langlois[2] , and Yvon Savaria[1]
[1]Dept. of Electrical Engineering, and [2]Dept. of Computer Engineering, École Polytechnique de Montréal
P.O. Box 6079 Station Centre-Ville, Montréal, QC, Canada H3C 3A7

***Abstract*: In this paper we propose a new deinterlacing algorithm using motion compensation and directional interpolation. To limit the propagation error that is a major drawback of conventional motion compensated methods, motion estimation is performed using original lines only, for same and opposite parity fields. In addition, a threshold value is used during the search to recognize situations where the motion estimator fails to find an optimal matching block. Enhanced edge-based line average with median filtering is used in these situations. Experimental results show that the proposed method performs better than the traditional motion compensated method, based on objective and subjective criteria.**

I. INTRODUCTION

Deinterlacing algorithms are used to improve the quality of interlaced signals adapted to enhanced digital television (EDTV) or high definition television (HDTV). Deinterlacers use spatial or temporal interpolation techniques to calculate the missing lines. Spatial interpolators do not improve the vertical resolution of the images and they sometimes blur them. Temporal deinterlacers attempt to interpolate missing lines based on temporal correlation. Temporal interpolation can improve the vertical resolution in static areas but it may produce some artifacts in moving areas.

Deinterlacing methods can be classified on the basis of whether they use motion compensation (MC) or not [1]. Within the non-motion compensated methods, the edge-based line average (ELA) method is widely used because it involves simple calculations and provides satisfactory results [2].

Motion compensated methods are known to be the best deinterlacing algorithms, but they have high computational complexity [3]. These methods use data of some neighbouring fields to detect moving objects or blocks. They calculate motion vectors for all blocks in the current field in order to calculate the missing pixels by temporal interpolation along the motion vectors. Motion vectors determine the displacement between blocks in the current field and the most similar blocks in neighbouring fields. These methods are highly dependent on the accuracy of motion estimation [4]. The selection of a robust motion estimation method is one of the most challenging issues in motion compensated methods.

Motion compensated methods attempt converting a moving block to a static one to mitigate production of artifacts by temporal interpolation [1]. These methods are more complicated for interlaced video data. Conventional motion compensated methods use some pre-filtering methods, such as line averaging, to calculate the missing pixels of the first field prior to motion estimation between opposite parity fields [4]. Therefore, this kind of motion estimation uses original as well as calculated pixels. This often propagates the error to subsequent fields, which is the most important drawback of all recursive methods, including motion compensated methods using opposite parity fields [1].

This paper is organized as follows. Section II explains the enhanced ELA and global motion compensated deinterlacing method (GMC). Section III presents the proposed deinterlacing algorithm. Section IV shows experimental results and a comparison with previous methods. Section V concludes the paper.

II. OVERVIEW

A .Enhanced ELA

Edge-based algorithms provide the best results within the class of intra field deinterlacing methods. These methods try to find the edge direction in the missing pixel's position and then they interpolate the missing pixels along that edge. Edge-based line averaging (ELA) uses three pixels in the upper and lower lines to detect the edge direction and perform directional interpolation. Modified ELA uses five pixels in the upper and lower lines to detect five edge directions. This method classifies the edges as dominant or non-dominant. It then uses directional interpolation in dominant edges and vertical interpolation in non-dominant edges. Enhanced ELA uses the same algorithm as the modified ELA to detect the dominant edges. Then it uses the median of the directional interpolated pixel and corresponding pixels in the upper and lower lines to prevent the occurrence of bursting pixels [5]. Fig. 1 shows the pattern of pixels in modified and enhanced ELA.

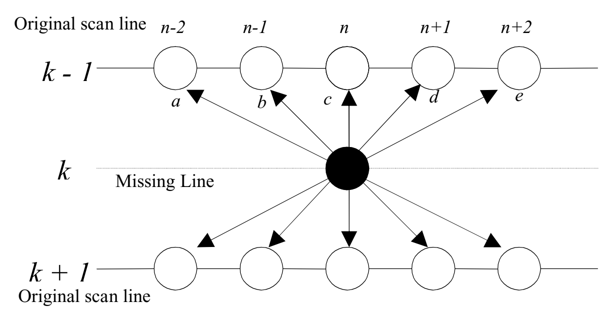

Fig. 1. Pattern of pixels used in Modified and Enhanced ELA

B. Global Motion Compensated (GMC) method

In traditional motion compensated methods, each field is partitioned in blocks, usually of size 8 by 8. For each block in the current field, a motion estimator then uses a block-matching algorithm to find the most similar block in the previous field. This block is called the "optimal matching block" (OMB). Motion vectors determine the displacement between blocks in the current field and OMBs in the neighbouring fields. Motion compensated methods use motion vectors to calculate the missing lines by temporal interpolation along the motion vectors.

Block matching methods have the highest computational complexity in motion compensated deinterlacers. They calculate the absolute cross reference error between all pixels in the current field block and matching blocks in neighbouring fields within a search window. The full search method provides the best results, but it has the highest computational complexity, and it is very difficult to implement in real-time systems. In this method, each block is compared with all possible blocks in the previous field within the search window.

The majority of fast block-matching methods are based on the assumption that the block matching error monotonously increases with the increase of the distance between the matching block and the optimal matching block. This assumption is not always true, so these methods may converge into a suboptimal solution [6].

In some motion compensated methods, prior to performing motion estimation for the first time, some pre-filtering algorithm such as line averaging is used to calculate the missing lines in the first field. The previously deinterlaced frame is then used by the motion estimator to calculate motion vectors for the next fields. Motion estimation then depends on interpolated pixels as well as original pixels in the previous field. Thus, any interpolation error may be propagated into the subsequent output frames. This is a common drawback of all recursive approaches including motion compensation methods. Some approaches such as five-point median filtering are used to prevent error propagation [4]. Although this is a very effective method, the median filter can introduce alias in the deinterlaced image [1].

After calculating the motion vectors, the missing lines of the blocks in the current field are replaced by the corresponding lines of the OMB in the previous field as follows:

$$F_o(x,y,n)=\begin{cases}F_i(x,y,n), & (y \mod 2 = n \mod 2)\\ F_o(x-d(x),y-d(y),n-1), & (otherwise)\end{cases} \quad (1)$$

where F_i is the input field, F_o is the output frame, $d(x, y)$ is the motion vector and n refers to the frame number.

III. PROPOSED DEINTERLACING ALGORITHM

Interlaced video data complicates motion compensated methods. Motion estimation may fail in interlaced data due to alteration of existing lines between odd and even fields. Even if the motion estimator successfully calculates the motion vector, how to calculate the missing lines remains an important issue in interlaced video data.

A. Motion Estimator

To overcome the problem of interlaced data, some MC methods use pre-filtering methods to deinterlace previous fields used by a motion estimator.

In the proposed method, the motion estimator uses only original lines for block-matching. This method is based on vertical motion within two sequential fields. Different values of vertical motion in the same and opposite parity fields are considered as corresponding pixels in opposite parity fields imply a different vertical movement that in same parity fields. A similar approach in the horizontal direction is not required, since horizontal down sampling is not present and the OMB displacement is independent of the parity of compared fields.

The full search method is used to find the OMB in the previous fields of same and opposite parities. A block size of 8 × 8 and a search window of 32 × 32 are selected. The motion estimator compares existing lines of the blocks in the current field with original lines only of blocks in the previous fields of same and opposite parities within the search window. Blocks from the set of all possible shifted blocks in the previous fields have a minimum displacement of two pixels in the vertical direction and one pixel in the horizontal direction. Due to the use of original lines only by the proposed motion estimator, the full search method has half the computational complexity when compared to the full search method in traditional motion compensation with pre-filtering.

We assume that the vertical motion within two sequential fields for each block is an integer number. Fig. 2 shows examples of various vertical motions for a block of 8 × 8 containing the capital letter "A" within the previous same and opposite parity fields. Two cases of odd and even vertical movements within three consecutive fields are considered.

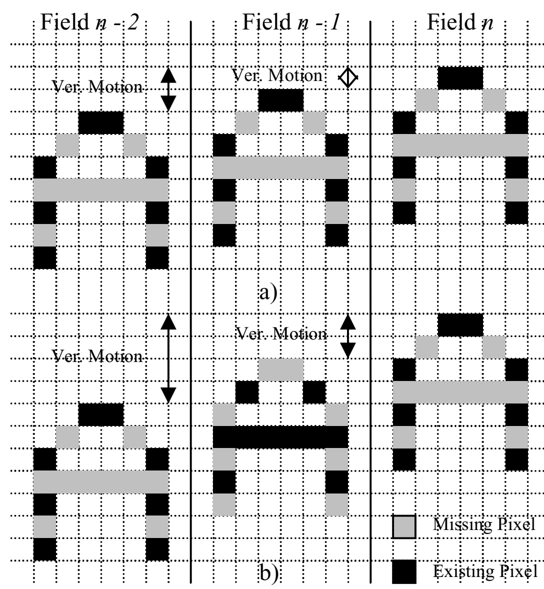

Fig. 2. a) Vertical Motion is an odd number
b) Vertical Motion is an even number

If the vertical motion within two sequential fields is an odd number, the existing and missing lines in the current field are also existing and missing, respectively, in the previous fields. Fig. 2-a shows the letter 'A' in three sequential fields with vertical movement of one pixel per field. The motion estimator can successfully find the optimal matching block in the previous fields because the existing lines of letter 'A' in the current field and the previous fields are the same. Now if the OMB is found in the previous field of the same parity, the vertical motion vector is even, and if it is found in opposite parity field it is odd.

In the next case, if the vertical motion within two sequential fields is an even value, the existing lines in the current field are missing in the previous field, but they are present in the previous field of the same parity. Fig. 2-b shows an example with vertical motion of two pixels within every two sequential fields. In this case, the motion estimator finds the optimal matching block in the previous same parity field. As it is shown, the existing lines of the letter 'A' in the current field are the same as the existing lines in the previous field of the same parity. Therefore, the motion estimator will find an OMB in the previous field of the same parity with vertical motion of an even number of pixels.

As the OMB is found for each block, the motion vectors are easily calculated by the displacement of each block and its corresponding OMB. According to our assumption that vertical movement in two sequential fields is an integer number, if OMB is found in the opposite parity field, the vertical motion is always an odd number and if it is found in the previous same parity field, it is always an even number.

B. Missing Lines Calculation

There are challenging circumstances for which traditional motion compensation methods produce artifacts or do not work properly [6]. These situations include: video sequences with rotation and scaling, when objects pass in front of one another, if an object moves beyond the motion vector search area, and finally if the objects or blocks have sub-pixel motion. In such cases, motion estimators may fail to find an OMB properly.

These problems lead us to come up with a criterion to check if the motion estimator has successfully found an OMB or not. This is done by comparing the OMB error to a threshold value determined empirically. If the OMB error is less than the threshold, an OMB has been found successfully. The OMB error is defined as follows:

$$Error(OMB) = \sum_{x=1}^{Xblksize} \sum_{y=1}^{yblksize} \left| F_i(x, y, n) - F_i(x + d(x), y + d(y), m) \right| \quad (2)$$

Where x and y point to the existing pixels in a block in the current field, $x + d(x)$ and $y + d(y)$ refer to OMB pixels, and m refers to the field where the OMB was found.

Different sequences were used to determine a good threshold value. This is determined by the pixel threshold multiplied by the number of pixels in a block.

Fig. 3 shows the flowchart of the proposed method. In the first step, the motion estimator searches the two previous fields to find the OMB. The estimator returns the motion vectors and the amount of error for OMB. If the error is greater than the threshold, the motion estimator has not successfully found the OMB in two previous fields. In this case, enhanced ELA is used to calculate the missing lines for that particular block.

If the error of the OMB is less than the threshold and the OMB is in the previous field (field $n - 1$), then the vertical motion vector is an odd number and the missing lines are not present in the previous fields. In this case, the missing lines will be replaced by the corresponding lines of the OMB that were deinterlaced previously.

If the OMB is found in the previous field of the same parity (field $n - 2$), the vertical motion is an even number. Now if the vertical motion is a number divisible by four, the missing lines of the block in the current field will be replaced with the original lines of the block with half motion vectors in field $n - 1$. Otherwise, the deinterlaced lines of OMB replace the missing lines of the current field block.

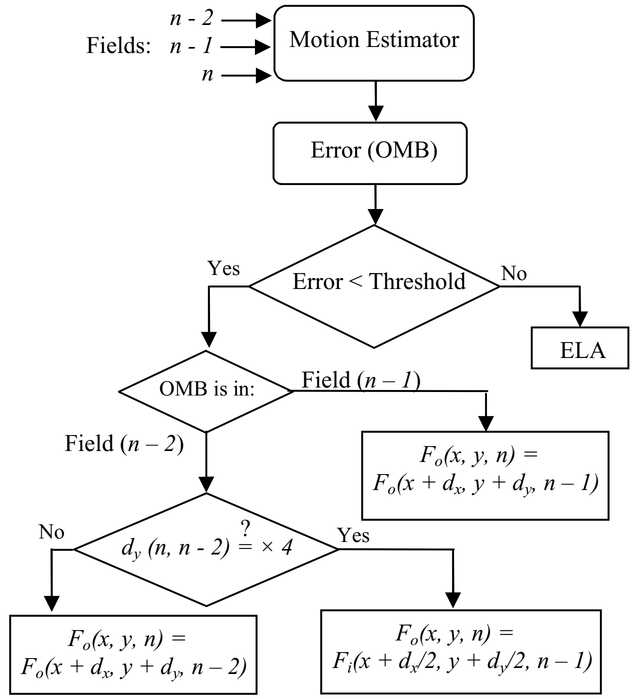

Fig. 3. Flowchart of proposed method

IV. ANALYSIS OF THE RESULTS

Four gray scale sequences were used to evaluate the new motion compensated method. Table 1 shows the average of the mean square error in 20 frames for different deinterlacing methods. A block size of 8 × 8, a search window size of 32 × 32, and the full search method were selected in both motion compensated methods. Prior to deinterlacing, line averaging is used for the first field in GMC and for the first two fields in the new method.

As it is shown in Table 1, the new method performs better than the other methods in all sequences except for the Foreman sequence deinterlaced by the Weave method. However, the line-crawling artifact, typical of the Weave method in moving areas, makes it a poor choice, even if it paradoxically exhibits a lower MSE.

Table 1. MSE comparison between different methods

Test Sequences	Line doubling	Line Ave.	Weave	ELA	GMC	Prop. Method
Baseball 160 × 112	506	269	509	326	677	259
Foreman 352 × 288	214	118	102	134	224	118
TableTennis 352 × 240	398	262	116	252	364	68
FlowerGarden 352 × 240	1135	754	1209	813	1087	318

Fig. 4 compares two motion compensated methods in some sequences of 20 frames. The proposed method achieves a lower

MSE and lower error propagation in all sequences. Fig. 5 also shows the deinterlaced images produced by the proposed and GMC methods for subjective comparison. The proposed method provides visually better quality than the GMC.

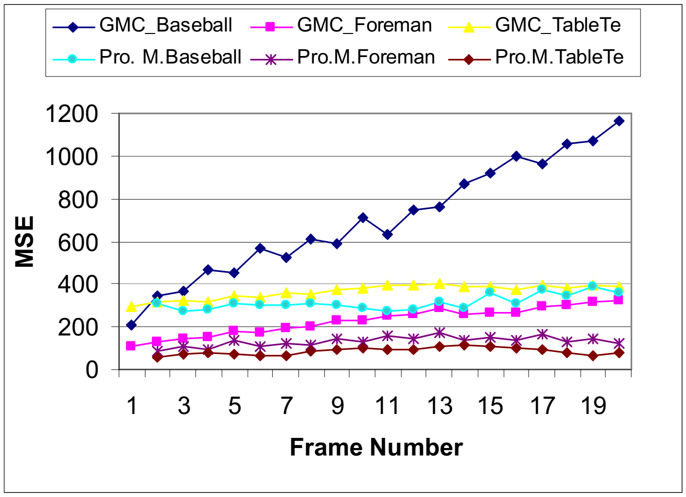

Fig. 4. MSE Comparison

V. CONCLUSION

A new motion compensated method was proposed. This method combines edge-based deinterlacing with motion compensation. The original lines in previous fields of the same and opposite parities are used by the motion estimator to calculate motion vectors. This limits the propagation error to the next frames and decreases the computational complexity of the full search method to half of the complexity of traditional method in opposite parity fields. Threshold comparison is used to detect situations where the motion estimator fails to find an optimal matching block. Spatial interpolation is performed with the enhanced ELA algorithm if motion estimation fails. Otherwise, motion compensation is used. Subjective and metric-based comparisons show that the proposed method compares favorably to previous approaches.

REFERENCES

[1] G. De Haan and E. B. Bellers, "Deinterlacing – An Overview," *Proc.of IEEE*, vol. 86, no. 9, pp.1839 – 1857, Sep. 1998.

[2] Sung-Gyu Lee and Dong-Ho Lee, "A motion-adaptive de-interlacing method using an efficient spatial and temporal interpolation," *IEEE Trans. on Consumer Electronics,* vol. 49, no. 4, pp. 1266-1271, Nov. 2003.

[3] E.B. Bellers and G. de Haan, "Advanced de-interlacing techniques*," Proc. ProRISC/IEEE Workshop on Circuits, Systems and Signal Processing*, Mierlo, Netherlands, pp. 1-13, Nov. 1996.

[4] Y.Y. Jung, B.T. Choi, Yung-Jun Park and Sung-Jea Ko, "An effective de-interlacing technique using motion compensated interpolation," *IEEE Trans. on Consumer Electronics*, vol. 46, no. 3, pp. 460-466, Aug. 2000.

[5] Shyh-Feng Lin, Yu-Lin Chang, and Liang-Gee Chen, "Motion adaptive interpolation with horizontal motion detection for deinterlacing," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1256-1265, Nov. 2003.

[6] Xinbo Gao, Juxia Gu, Jie Li, "De-interlacing Algorithms Based on Motion Compensation," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 2, pp. 589-599, May 2005.

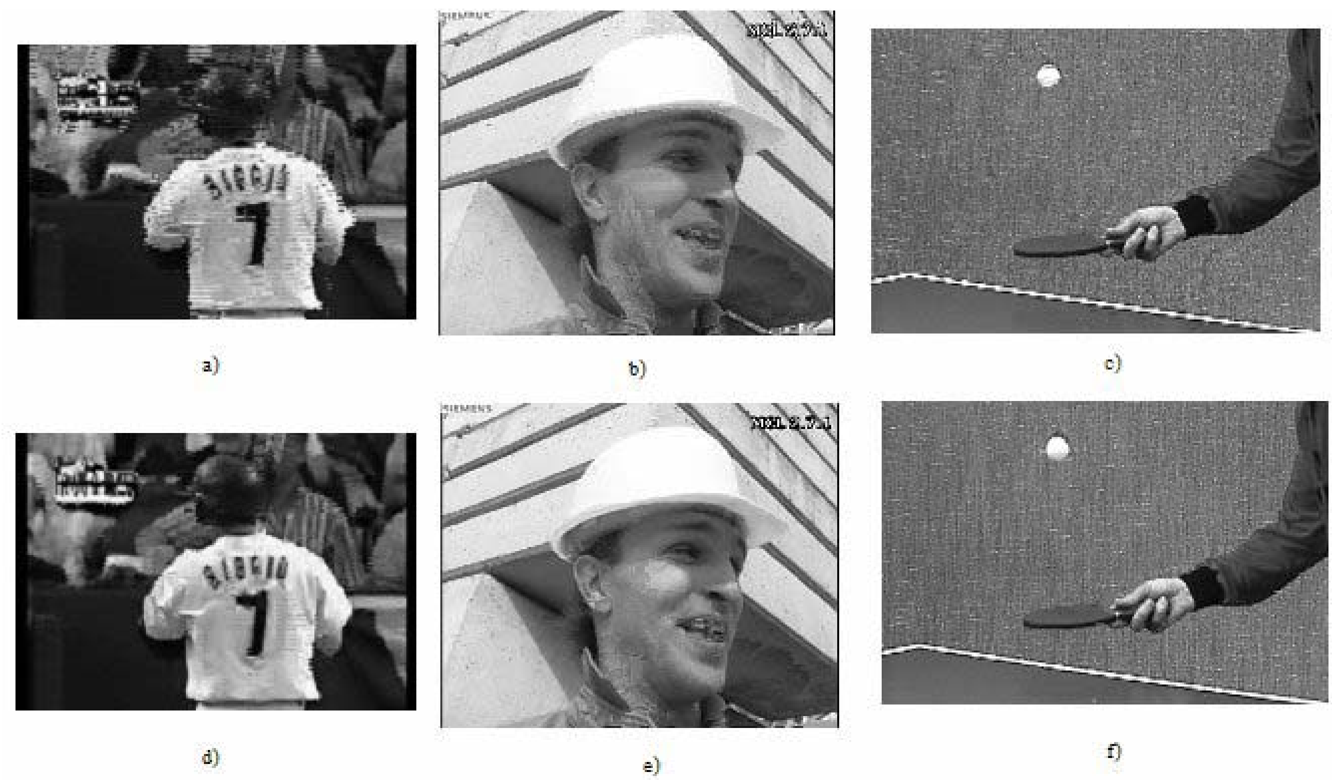

Fig. 5. Up: GMC Down: Proposed Method a), d) Baseball b), e) Foreman c), f) Table_Tennis