Hybrid Deinterlacing Algorithm with Motion Vector Smoothing.

*Khvan, Dmitriy **Jeon, Gwanggil ***Jeong, Jechang
Hanyang University
**University of Incheon
*dm_kvn@hotmail.com

Abstract

In this paper we propose a new deinterlacing method with block classification and motion vector smoothing. The proposed method classifies a block, then depending on the region it belongs to, the motion estimation or line averaging is applied. To classify a block its variance is calculated. Then, for those blocks that belong to simple non-texture region the line averaging is done while motion estimation is applied to complex region. The motion vector field is smoothed using median filter what yields more accurate interpolation.

In the experiments for the subjective evaluation, the proposed method has shown satisfying results in terms of computation time consumption and peak signal-to-noise ratio. Due to the simplicity of the algorithm computation time was drastically decreased.

1. Introduction

Interlacing is a procedure of down sampling where vertical sampling rate is halved. Interlaced video was a popular method of bandwidth compression back in times when analog television was used. Interlaced scan refers to reducing vertical resolution of a frame by eliminating odd or even lines of a picture. As the Nyquist sampling theorem states, if the spatial vertical frequency of the original image is less than half of the maximum spatial frequency corresponding to progressive scan, then the interlaced signal can theoretically be reconstructed by intra-field deinterlacing [1].

To display interlaced video on a progressive screen, the resolution has to be restored, e.g. missing pixels have to be interpolated. First successful works in deinterlacing area took place in the late 80's, when progressive scan screens came into being. At that time two main groups of deinterlacing methods were formed: intra-field deinterlacing and inter-field deinterlacing.

Intra-field deinterlacing refers to restoring missing pixels using inner frame pels. Intra techniques take advantage of inner frame pixel correlation. In turn, inter-field deinterlacing performs motion estimation in order to interpolate missing pixels with pixels of a reference frame. This requires certain amount of computation what makes intra-deinterlacing preferable due to its simplicity. Intra-deinterlacing methods are fast and effective. However, the advantage of inter-deinterlacing algorithms is that using motion estimation we can possibly interpolate missing lines of a current frame with original lines of a reference frame. In this case, interpolation error may be reduced compared to intra-deinterlacing.

Deinterlacing algorithms deal with the problem of reducing the artifacts associated with interlaced video. Among those visually annoying artifacts the most common one is flickering, in other words interpolated line displacement in relation to original lines.

To deal with problems associated with deinterlacing and flickering artifact in particular, several authors propose to use intra-field methods together with inter-field algorithms. Hybrid deinterlacing algorithms have combined intra-field and motion-compensated methods based on reliability of calculated motion vectors. However, whether these algorithms are powerful enough to produce progressive scan video sequence is not evident.

Currently, we have a broad choice of deinterlacing methods. In [1] an overview of deinterlacing algorithms is presented. The overview provides no final answer in the interlace debate. The question "which algorithm is optimal" is still left unanswered. However, deep analysis of existing technical and non-technical issues gives us an idea of what algorithm we would prefer. We are given liberty in balancing between quality and complexity, which is the main challenge of the whole image processing area.

This paper is organized as follows. In section 2, the proposed algorithm is described in details. Section 3 presents experimental results. Section 4 concludes the paper.

2. Proposed method

The proposed deinterlacing algorithm combines intra and inter deinterlacing techniques. In intra deinterlacing, the missing lines are interpolated using simple line averaging. In turn, inter deinterlacing is presented by full search motion estimation. Several other works such as [2],[3],[4] use line averaging as well as motion estimation to deinterlace one frame. However, this paper has a different approach when judging whether to interpolate the missing lines using line averaging or to perform motion estimation. In [2] where 5 fields are considered for one frame deinterlacing, if OMB (Optimal matching block) is found in the opposite-parity field, the line averaging is performed. In [3] written by the same authors, inter-intra decision is based on motion vector reliability. In both cases motion estimation has to be done prior to inter-intra decision. In this paper, we firstly classify a block, in other words we decide whether it belongs to high-complex region or to low-complex background. Then, motion estimation is done for high-complex region, while line averaging is done for low-complex background. This considerably reduces computational time required.

The flow chart of the algorithm is presented in Figure 1.

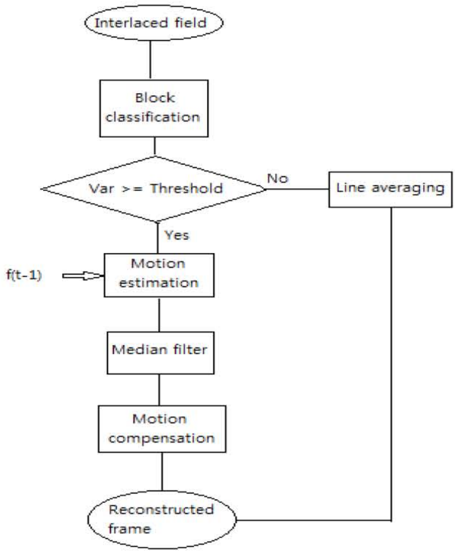

Figure 1. Block diagram of the proposed algorithm.

a. Block classification

In this paper the block variance is used as a criterion for inter-intra decision. The variance of a block is found using formulas (1) and (2)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$Var = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{2}$$

where n - number of pixels in a block, x_i - pixel intensity value, μ - average of pixels in a block.

Then, comparing the variance of a block with the empirically calculated threshold, a frame is divided into blocks that will be interpolated with line averaging (intra-blocks) and those that are to be interpolated using motion estimation (inter-blocks).

Depending on a video sequence the ratio of intra-blocks to inter-blocks may vary. We have adjusted the ration to be optimal for most of sequences by setting the threshold to be equal 3000.

b. Motion estimation and compensation

The basic block matching algorithm (BMA) scheme is shown in Figure 2. Each block of the current frame is compared with the co-located block in the previous frame to yield motion vector, or relative displacement along rows and columns.

To further decrease computation time, other methods that are faster than the full search method may be used. However, these methods decrease the accuracy of motion vectors. The motion vector fidelity is crucial in the algorithm. Mainly for this reason the full search algorithm will be used.

Compared to several earlier works in the deinterlacing area, the motion estimation step was slightly modified. First of all, instead of using 4 reference frames for one current frame interpolation, we use just only one previous frame as the reference frame. Secondly, we do not apply line averaging prior to motion estimation as it is done in [2]. In [2], if a pixel is wrongly interpolated using line averaging this error will propagate after motion estimation. Moreover, motion estimation does not always give better result than line averaging. In summary, we concluded that it is not justified to apply line averaging before motion estimation. For this reason only original pixels are used for motion estimation as shown in Figure 3.

$$V = \arg\min_{v \in S} \sum_{x \in B} \left| f(x+v, n-1) - f(x, n) \right| \tag{3}$$

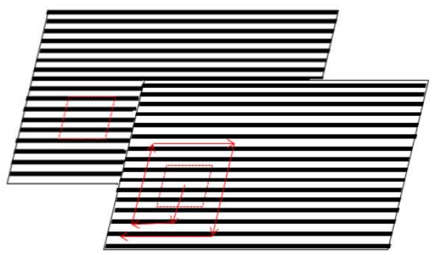

Figure 2. Spiral search block matching algorithm.

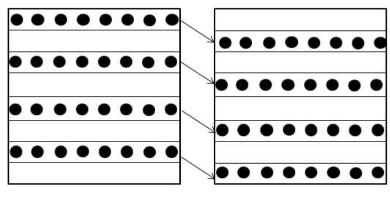

Figure 3. Motion estimation using only original lines.

c. Line averaging

A simple intra deinterlacing method such as line doubling or line averaging with time has evolved to extremely complex algorithms. These algorithms involve extensive mathematical computation and will not be used in the proposed algorithm, because we set a goal of reducing the time consumption.

d. Median filter

To increase motion vector reliability motion vector filtering is applied. Median filter is defined as (4).

$$\sum_{i=1}^{N}\left\|\vec{v}_{WVM}-\vec{v}_i\right\|_p \le \sum_{i=1}^{N}\left\|\vec{v}_j-\vec{v}_i\right\|_p,\ j=1,2,\ldots,N \quad (4)$$

Median filters for motion vectors are quite effective in reducing impulse noise whilst preserving contours [5]. The reliability is improved by eliminating erroneous motion vectors (motion outliers). However, since a frame consists of intra-blocks as well as inter-blocks, in other words, a motion vector field does not cover the whole frame, not all motion vectors can be smoothed. If a block has 8 neighbors in 3x3 window then filtering is applied.

3. Experimental results

For the experiment three Quarter Common Intermediate Format (CIF: 352x288) sequences, namely: Football, Coastguard and Children were used. The threshold for block classification is empirically set to 3000.

In our experiments, we evaluated the performance of the proposed hybrid deinterlacing algorithm in two respects. First, to evaluate the visual quality we present a chart of PSNR results for several algorithms including the proposed one. Secondly, in order to demonstrate the performance of the proposed algorithm in terms of complexity, the chart of time consumption is presented.

Table 1. PSNR results.

Sequence	Algorithm	PSNR (db)
Football 150 frames	Line Averaging	26.13
	FFMC	26.19
	RME	27.80
	Proposed	26.58
Sequence	Algorithm	PSNR (db)
Coastguard 300 frames	Line Averaging	24.20
	FFMC	23.12
	RME	26.16
	Proposed	24.79
Sequence	Algorithm	PSNR (db)
Children 300 frames	Line Averaging	25.65
	FFMC	25.89
	RME	27.13
	Proposed	26.57

Table 2. CPU time results.

Sequence	Algorithm	CPU Time (sec)
Football 150 frames	Line Averaging	1.3
	FFMC	81.3
	RME	209.6
	Proposed	76.6
Sequence	Algorithm	CPU Time (sec)
Coastguard 300 frames	Line Averaging	3.7
	FFMC	257.3
	RME	411.7
	Proposed	249.1
Sequence	Algorithm	CPU Time (sec)
Children 300 frames	Line Averaging	3.1
	FFMC	247.1
	RME	434.2
	Proposed	223.6

Figure 4. FFMC algorithm.

Figure 5. RME algorithm.

Figure 6. Proposed algorithm.

4. Conclusion

To conclude the paper we make an analysis of PSNR results. As it can be seen from the PSNR table the proposed algorithm does not perform better than RME algorithm. However the proposed algorithm is much faster. In fact, for particular applications simplicity of a algorithm is more valuable than PSNR. On top of that, the PSNR does not always give adequate visual quality evaluation.

Acknowledgement

This work was supported by the Industrial Strategic technology development program(10039886, Combined Modulator and Demodulator with Real Time Multi Codec Transcoder based on Hardware Processing) funded by the Ministry of Knowledge Economy(MKE, Korea).

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] H.M. Mohammadi, P. Langlois, Y. Savaria, "A Five-Field Motion Compensated Deinterlacing Method Based on Vertical Motion," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 3, pp.1117 - 1124, Aug. 2007.

[3] H.M. Mohammadi, P. Langlois, Y. Savaria, "Hybrid video deinterlacing algorithm exploiting reverse motion estimation," *IET Image Processing*, vol. 5, no. 7, pp. 611-618. Oct. 2011.

[4] Yu-Cheng Fan, Hung-Shin Lin, Chiang A., Hen-Wai Tsao, Ching-Chih Kuo, "Motion Compensated Deinterlacing With Efficient Artifact Detection for Digital Television Displays," *Display Technology*, vol. 4, no. 2, pp. 218-228, June 2008.

[5] L. Alparone, M.Barni, F. Bartolini, V. Cappelini, "Adaptively weighted vector-median filters for motion-fields smoothing," *IEEE Trans. on Consumer Electronics*, vol. 4, pp. 2267-2270, May 1996.

[6] You-Young Jung, Byung-Tae Choi, Yung-Jun Park and Sung-Jea Ko, "An Effective De-interlacing Technique Using Motion Compensated Interpolation," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 460-466, Aug. 2000,

[7] K. Sugiyama and H. Nakamura, "A method of de-interlacing with motion compensated interpolation," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 611 - 616, August 1999.

[8]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.

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

[10] Jelena Kovacevic, Robert J. Safranek and Edmund M. Yeh, "Deinterlacing by Successive Approximation," *IEEE Transactions on Image Processing*, vol. 6, no. 2 , pp. 339-344, Feb. 1997.