AN ADAPTIVE AND PARALLEL SCHEME FOR HD VIDEO DE-INTERLACING

Jie Dong and King Ngi Ngan

Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR

ABSTRACT

It is very challenging to de-interlace HD videos in real time, as both high efficiency and low complexity should be fulfilled, which, however, are conflicting. This paper presents a de-interlacer to resolve the conflict specially for H.264 coded videos. It adapts to spatially and temporally local activities by making full use of the syntax element (SE) values in bit-streams, which give many hints of the motions and textures of video sequences. Accuracy analysis is also introduced to deal with the disparity between the SE values and the real motions and textures. The experimental results show the proposed de-interlacer provides better visual quality than common ones and can de-interlace 1080*i* sequences in real time on PCs.

1. INTRODUCTION

Nowadays, High-Definition (HD) videos become more and more popular with many applications, e.g., HDTV broadcasting and high-density storage media. Among the HD video formats, 1080*i* is the most widely used and has been adopted by many countries for HDTV broadcasting. 1080*i* is directly compatible with CRT-based HDTV sets, which, however, never entered the market in large volume, and is also compatible with 1080*p*-based HDTV, if de-interlacing is performed immediately before display. In other words, every field in 1080*i* videos should be converted to a frame in real time.

It is very challenging to de-interlace HD videos in real time, especially by using software. The algorithm should be simple on one hand, to process such a large amount of data within the limited time, and be flexible on the other hand, to adapt to spatially and temporally local activities. The latter aspect inevitably employs implicit or explicit motion and texture detectors, which makes the algorithm more complicated and conflicts with the former aspect.

Most of the existing de-interlacing techniques aim at one aspect. Some basic methods are simple enough to fulfill the real-time requirement at the expense of visual quality, including various linear and median filters (MF). Regardless of time constraint, e.g., in the filming studio, some advanced techniques [1][2] can provide good visual quality by employing motion compensation (MCP) methods [3], which are adaptively used with basic methods according to motion detection and estimation. Recently, some research work took both aspects into consideration and tried to resolve the conflicts [4][5][6], but never reported the computational time. [4] is based on MCP method and reconstructs a frame in an iterative way, and [5] uses the hybrid of motion detector and extended MF, both of which are still too complicated to achieve real-time implementation for Standard-Definition (SD) videos, let alone HD videos. [6] can de-interlace HD videos in real time, but introduces obvious serrations.

To resolve the conflicts, this paper proposes a de-interlacer specially for H.264 coded videos, which can de-interlaces 1080*i* videos in real time and provides good visual quality. Firstly, the proposed de-interlacer makes full use of the SE values during the decoding process, as if they are the output of complicated motion or texture detectors, thus saving the time for motion and texture detection. Secondly, accuracy analysis is introduced to deal with the disparity between the SE values and the real motions and textures, caused by the encoding strategies, and only the reliable SE values are used to choose local interpolation methods. Furthermore, this de-interlacer is designed for parallel implementation, so can take advantage of multi-core systems, which are common for PCs and DSP platforms nowadays.

The remainder of the paper is organized as follows. Section 2 introduces the proposed real-time de-interlacer in detail. Section 3 gives the experimental results, followed by the conclusion in Section 4.

2. THE PROPOSED REAL-TIME DE-INTERLACER

With the proposed de-interlacer, the field to be de-interlaced is first divided into blocks, called process unit, in which the missing pixels are interpolated by the same method. The size of each process unit is the area of the field cropped by the associated MB, i.e., 16×16 for field-coded MBs and 16×8 for frame-coded MBs. The process of de-interlacing one field consists of two stages. The first stage, named decision stage, analyzes the reliability of the SE values, decides the interpolation method for each process unit, and output a 2-D mode map recording the interpolation methods of all the process units in one field. The second stage, named interpolation stage, applies the appropriate interpolation methods to the process

The work was supported in part by the Hong Kong Applied Science and Technology Research Institute (ASTRI) and the Chinese University of Hong Kong Focused Investment Scheme (Project 1903003).

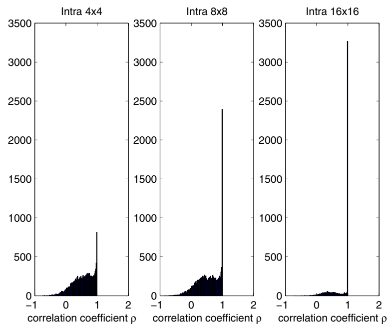


Fig. 1. The histogram of ρ of vertically neighbouring pixels in intra-coded MBs (based on sequence *StockholmPan*)

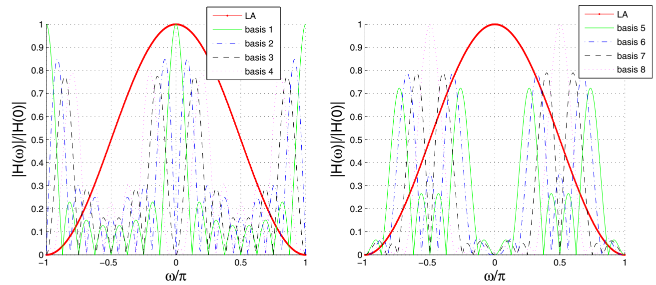
units, as indicated in the 2-D mode map. The two stages are introduced detailedly in Section 2.1 and 2.2, respectively.

2.1. Decision stage

With the proposed de-interlacer, the interpolation method of each process unit is determined by the SE values of the associated MB. Therefore, the texture smoothness of intra-coded MBs and the proportion of different MB partitions are investigated at first, and then the flow charts for mode decision are designed accordingly as below.

If the MB is intra-coded, the correlation coefficient ρ of the vertically neighbouring pixels in this MB is calculated, where each column is regarded as an observation of a discrete stochastic process and each row is a random variable. The results show that intra16×16-coded MBs always indicate very smooth areas with vertical correlation approaching 1.0, whereas the pixels in intra4×4-coded MBs have smaller correlations, representing fine textures. For these two cases respectively, line averaging (LA), which is simple and efficient for areas without vertical aliasing, and edge-based line averaging (ELA) [7], which has outstanding performance for sharp and consistent edges, are applied for the associated process units. For the intra8×8-coded MBs, the textures have almost the equal probability to be fine and smooth. As an example of this observation, Fig. 1 shows the histogram of ρ of vertically neighbouring pixels in different intra-coded cases, which is obtained by investigating *StockholmPan*.

When determining the interpolation method for the process unit associated with intra8×8-coded MB, we first check the condition that all the 8×8 blocks in the MB are vertically smooth. In detail, for each 8×8 block, let $f(i, j)$, $0 \le i, j < 8$, be the transform coefficient in the position of the i^{th} row and the j^{th} column, and if all $f(i, j)$, $i > 3$, are equal to 0, the 8×8 block is considered to be vertically smooth. LA is suitable for de-interlacing vertically smooth 8×8 blocks, because vertically, such blocks are represented by only the first four bases of the 8×8 transform matrix in H.264 and the spectra of these bases are plotted in Fig. 2 (a), of which the main lobes are all in the pass-band of the LA filter. However, the main lobes of the spectra of the other four bases are in the



(a) (b)

Fig. 2. Frequency response of LA and the spectra of bases

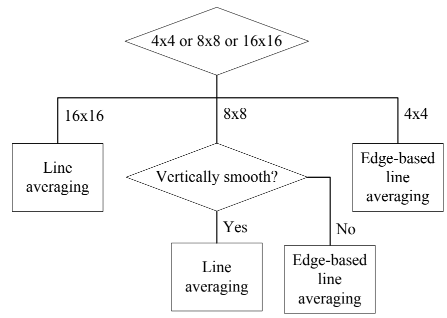


Fig. 3. The mode decision for intra-coded process unit

stop-band (see Fig. 2 (b)) of LA, so non-linear filter ELA will be applied to blocks not vertically smooth. In short, the proposed decision flow for the process unit associated with intra-coded MB is shown in Fig. 3.

While de-interlacing intra-coded pictures, the interpolation method switches between LA and ELA, which, however, is at the mercy of aliasing. Aliasing, such as flicker and crawling lines, can be observed when the de-interlaced frames are successively displayed. Therefore, if the video sequence is all-intra-coded, the defect caused by the proposed decision flow (see Fig. 3) becomes obvious. Fortunately, for most HD applications, the interval of I pictures is no smaller than 0.5 second. This drawback will not influence the visual quality much, as proved by the subjective test, and can be overcome by MCP methods involved in the inter-coded pictures.

For the case of inter-coded MB, the proportion of different MB types is investigated. Table 1 shows the results for B frames at the bit-rate about 15Mbit/s, i.e., PSNR around 38dB. The proportion of skip and direct modes is often larger than 50% for HD videos. With the two modes, the motion vectors (MV) are derived from those of spatially or temporally neighbouring blocks and thus indicate uniform translation. Hence, MCP is a safe method for the associated process unit. Otherwise, the MB partition is checked as follows.

If it is not larger than 8×8, which means rich details inside and the MVs are very likely to be inaccurate, intra-field method, i.e., ELA, is applied. Otherwise, the MB partition is larger than 8×8, where there is the possibility of either smooth or complicated motions and textures. MCP method will be applied, if the residues are transformed by the 8×8 ICT and vertically smooth. Otherwise, the ELA will be ap-

Table 1. The percentage of different MB types in B frames

Test sequence	Skip/Direct	16×16	16×8	8×16	8×8	Intra
Parkrun	69.09	19.11	3.07	1.95	6.74	0.04
Shields	68.81	16.49	3.35	2.62	8.57	0.21
StockholmPan	71.22	13.70	3.76	2.85	8.43	0.09

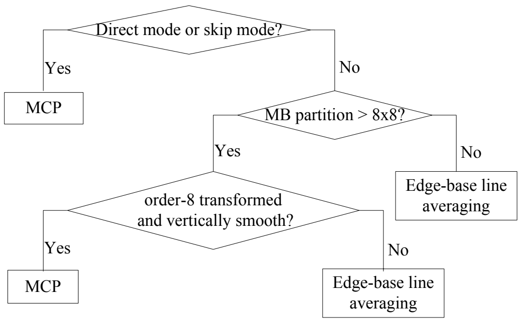


Fig. 4. The mode decision for inter-coded process unit

plied. In short, the proposed decision flow for the process unit associated with inter-coded MB is shown in Fig. 4.

Besides Fig. 3 and Fig. 4, a complementary MCP (CMCP) method is introduced for the frames that are far from their reference frames, e.g., hierarchical B frames and P frames in IBBPBBP sequence structures. Taking the IBBPBBP sequence as an example, the percentage of skip modes in P frames is greatly reduced (see Table 2), as uniform translation is not likely to continue over such a long interval, at least 6 fields. Actually, most MBs in P frames have accurate matching blocks in the temporally neighbouring non-referred frames in the display order. The CMCP allows the process units in P frames to find the best matching blocks in the neighbouring non-referred frames, as presented in Section 2.2.

2.2. Interpolation stage

In this stage, each process unit in a field is interpolated using the corresponding method in the 2-D mode map, the output of the decision stage. The decision and interpolation stages can work in parallel in a multi-core system. The decision stage is implemented together with the reconstruction process in order to collect and analyze the SE values conveniently, so is in the decoding order. The interpolation stage is independent, so can be performed behind the decision stage in the display order. Furthermore, the field can be divided into parts to be interpolated in parallel according to the number of cores.

If interpolated by MCP method, the process unit has the same partition and directions of prediction as those of the associated MB. Only the immediately previous and next fields are used as reference for MCP method, regardless of the number of reference frames in H.264 bitstreams. The MVs pointing to other fields are first linearly scaled to the neighbouring fields. If the pixels in a neighboring field, to which the MV points, are not on the interlaced sampling grid, temporal backward projection (TBP) [8] is employed to solve the problem. TBP uses existing pixels for MCP instead of interpolated

Table 2. The percentage of different MB types in P frames

Test sequence	Skip	16×16	16×8	8×16	8×8	Intra
Parkrun	11.36	30.82	15.80	11.04	17.76	13.3
Shields	4.15	31.09	16.21	13.21	23.47	12.1
StockholmPan	7.07	22.53	14.46	11.41	24.73	19.92

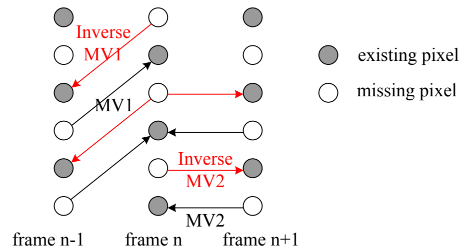


Fig. 5. The complementary MCP method

ones, which avoids error propagation, and is very simple to fulfill the real-time requirement. As the drawback, it implicitly assumes correct MVs. However, with the aforementioned mode decision process (see Fig. 4), most of the MCP methods are selected, when the associated MB are skip or direct more, which fulfills the assumption. Further protection is also introduced to improve the robustness against incorrect MVs. The pixels used for MCP are passed a 3-tap MF as shown in (1) instead of being used directly,

$$p_{i,n} = median(p_{i-1,n}, p_{i+1,n}, r) \qquad (1)$$

where $p_{i,n}$ is the pixel to be interpolated, $p_{i-1,n}$ and $p_{i+1,n}$ are the upper and bottom existing pixels, and r is the prediction value. In case the MCP is bi-directional, r in is the mean of the forward and backward pixels the MVs point to.

As the interpolation stage is performed in the display order, the aforementioned CMCP method becomes possible. As shown in Fig. 5, frame n is where CMCP applies to. If the existing pixels in frame n are used as the references by the process unit M in the neighbouring frame and the transform coefficients of M are all zero, which means the reference and target blocks perfectly match, the inverse MV will be used for MCP of the missing pixels in frame n. As the missing field of frame n and the existing fields of frame $n-1$ and $n+1$ have the same parity, the inverse MV always points to existing pixels in frame $n-1$ or $n+1$. If the reference pixels from frame $n-1$ and $n+1$ have overlapping in frame n, the mean will be used as the prediction value. The CMCP can increase the usage of MCP method (see Table 3) and thus reduce flicker.

3. EXPERIMENTAL RESULTS

3.1. Subjective quality

The proposed de-interlacer provides better visual quality than many other real-time de-interlacrs, including basic methods and [6]. It removes serrations, which often remain after temporal linear filtering, and eliminates flickers, while intra-field methods cannot. The large proportion of MCP method used during the proposed de-interlacing, i.e., 70% on average, contributes much to the flicker elimination. It also outperforms

Table 3. The percentage of MCP before and after CMCP

	Parkrun	Shields	StockholmPan
before CMCP	33.90	12.06	11.21
after CMCP	60.11	55.10	53.33

Table 4. The time (second) for de-interlacing 116 fields

Test sequence	IPP dec.	IPP dec.+De-interlacing	Computational time
Flamingo	4.354	4.478	0.124
Kayak	4.208	4.423	0.215
Mountain	4.426	4.639	0.213
Parkrun	4.335	4.375	0.040
Shields	4.109	4.697	0.588
Stockholmpan	4.687	4.929	0.242
Average	4.353	4.590	0.237

[6], a trade-off between flexibility and simplicity. With [6], serrations can still be observed and the edge blur is more severe, as shown in Fig. 6, where the images are clipped from 1080*i* de-interlaced frames and the sequences for de-interlacing are reconstructed from H.264 20Mbit/s bitstreams.

3.2. Computational time

The PC used to test the computational time has two 3.0 GHz Dual Core Intel Xeon CPUs and 1 GB RAM. The proposed de-interlacer is implemented to maximize the CPU usage of the four-core system, and integrated into the Integrated Performance Primitives (IPP) decoder, which decodes H.264 bitstreams by multi-threading also. Two Win32 functions QueryPerformanceFrequency and QueryPerformanceCounter are used for timing, which measure the execution time to nanosecond accuracy. Table 4 compares the average time of decoding 116 fields by the IPP decoder and decoding with de-interlacing 116 fields by the IPP decoder with the proposed de-interlacer integrated in. On average, the execution time of the proposed de-interlacer is only 0.237s for 116 fields, which means the de-interlacer occupies only 10% of the decoding time, if the speed for decoding 1080*i* is 50 fps, thus achieving real-time de-interlacing for HD videos.

4. CONCLUSION

This paper presents an efficient real-time de-interlacer specially for H.264 coded videos. It makes full use of SE values as the hints of motions and textures, but does not entirely relies on them, and is also suitable for parallel implementation. The experimental results prove this de-interlacer provides better visual quality than common ones and can de-interlace 1080*i* sequences in real time on PCs.

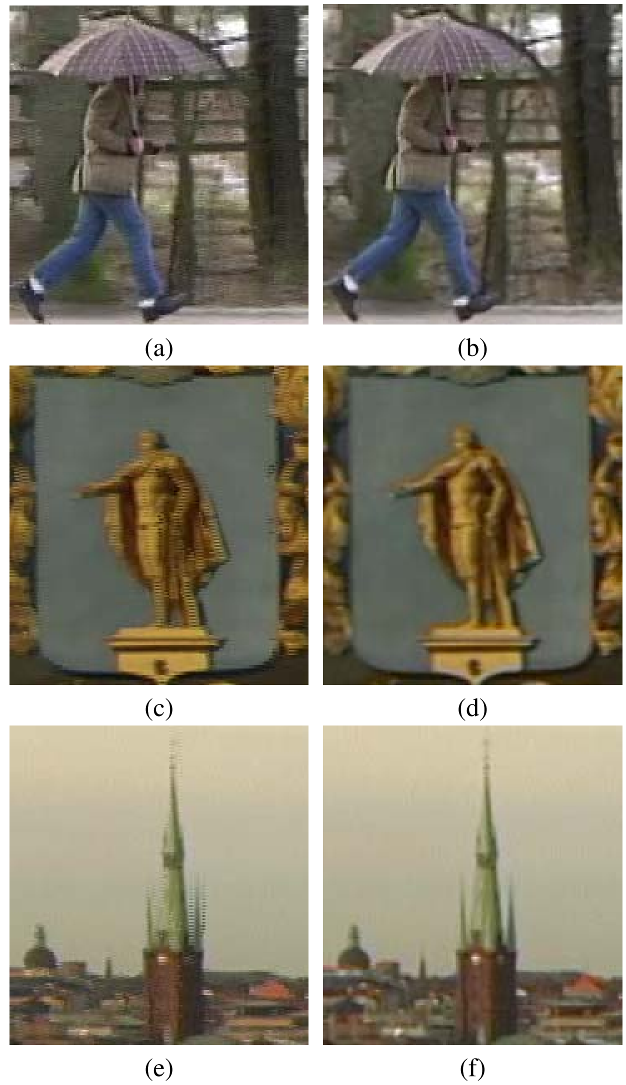


Fig. 6. The visual quality compared with [6]. (a), (c), and (e) are provided by [6], whereas (b), (d), and (f) are provided by the proposed de-interlacer

5. REFERENCES

[1] D. Wang, A. Vincent, and P. Blanchfield, "Hybrid de-interlacing algorithm based on motion vector reliability," *IEEE Trans. Circuits Syst. Video Technol.*, vol.15, pp.1019-1025, Aug. 2005.

[2] Y. L. Chang, et al., "Video de-interlacing by adaptive 4-field global/local motion compensation approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol.15, pp.1569-1582, Dec. 2005.

[3] G. de Haan and E. B. Bellers, "De-interlacing of video data," *IEEE Trans. Consum. Electron.*, vol.43, pp.819-825, Aug. 1997.

[4] D. Van De Ville, W. Philips, and I. Lemahieu, "Motion compensated de-interlacing for both real time video and still images," *IEEE Int. Conf. Image Process.*, vol.2, pp.680-683, Sept. 2000.

[5] Q. Huang, W. Gao, D. Zhao, and H. Sun, "Adaptive deinteralcing for real-time applications," *Pacific-Rim Conf. on Multimedia*, LNCS 3768, pp.550-560, 2005.

[6] M. J. Chen, C. H. Huang and C. T. Hsu, "Efficient de-interlacing technique by inter-field information," *IEEE Trans. Consum. Electron.*, vol.50, pp.1202-1208, Nov. 2004.

[7] T. Doyle and M. Looymans, "Progressive scan conversion using edge information," in *Signal Processing of HDTV II*, pp.711-721, Elsevier, 1990.

[8] J. W. Woods and S.-C. Han, "Hierarchical motion compensated de-interlacing," in *Proc. SPIE*, vol.1605, pp.805-810, 1991.