Algorithm Adaptive Video Deinterlacing Using Self-Validation Framework

Ting-Chun Wang [1], Yi-Nung Liu [2], Shao-Yi Chien [2]
[1] *Department of Electrical Engineering* [2] *Graduate Institute of Electronics Engineering*
National Taiwan University

Abstract—**Deinterlacing is well known as an ill-posed problem of video restoration. In this paper, a self-validation framework is proposed to solve the problem. First, various algorithms with different assumptions will be performed to generate candidate results. Then, a method called double interpolation is applied to test the consistency of each algorithm. Finally, the value of each missing pixel will be chosen among the results of those algorithms according to their performances in the previous test. By doing so, each situation can be handled by the most suitable algorithm and the overall result can be better than any of them. Experimental results demonstrate the validity of the proposed framework in both subjective and objective assessments.**

I. Introduction

Interlaced videos, as opposed to progressive videos, are scanned as a sequence of fields comprising alternating odd or even lines of a frame. By making videos interlaced, a trade-off between quality and bandwidth can be achieved, which is adopted by most broadcasted television systems. First designed for display on Cathode Ray Tube (CRT) systems, interlaced videos are still the most popular analogue video format now, even having a market share in the new high-definition standards such as HDTV-1080i. Nevertheless, interlaced videos suffer from visual artifacts such as serration, line crawling or flickering, which was not a problem in the past simply because the screen was small so the effect was not apparent. In addition, recent development of flat panel displays, e.g., LCD and PDP require progressive videos to be utilized, so there is an increasing demand for a method for converting interlaced videos to progressive ones. This process is commonly known as deinterlacing.

There are mainly two categories of deinterlacing algorithms: intra-field methods and inter-field methods. Intra-field methods, including line doubling, vertical spatial interpolation [1] and edge-based line averaging (ELA) methods [2], exploit information only within the current field to interpolate the missing pixels. Inter-field methods, on the other hand, consist of a family of motion compensation (MC) algorithms [3] and may utilize temporal information via motion estimation. Although the latter show more promising results in general, they also turn out to have higher complexity and are more error-prone when the motion estimation fails. Motion adaptive (MA) methods [4] use either an intra-field method or an inter-field method depending on its motion detection result.

Recently, a number of state-of-the-art algorithms have been proposed to better address the problem. Yu-Cheng Fan *et al.* [5] presented a motion compensation algorithm with efficient artifact detection to validate soft-switching between intra- and inter-field interpolations. The variational method proposed by Sune Høgild *et al.* [6] adopted a framework originally used in inpainting to derive an MA deinterlacer as well as an MC deinterlacer. Kwon Lee *et al.* [7] used a content adaptive vertical temporal filtering method to effectively utilize correlations between adjacent frames. The motion adaptive algorithm proposed by Hyunsoo Choi *et al.* [8] took advantage of modular neural network and used motion vectors to select optimal input pixels from neighboring fields.

Although all the above algorithms solve some of the problems, none of them is able to be free of artifacts entirely. Nevertheless, if there is an evaluation method that can judge how well an algorithm performs without having the ground truth in hand, those badly behaving algorithms can be avoided and the most appropriate one can be chosen. This is the main idea of self-validation.

The rest of this paper is organized as follows. The framework of self-validation process is introduced in Section II. Section III describes how to use the process to implement the proposed algorithm. Section IV shows the results and the comparison with other algorithms. Finally, the conclusion is presented in Section V.

II. Self-validation

All deinterlacing algorithms have some assumptions to work out their results, and they may all perform well in smooth regions. However, in other cases which do not match their hypotheses, their performances may not be that good. Given that all of them can deal with some situations, if a method could be utilized to judge whether an algorithm suits the current condition without knowing the ground truth, the most appropriate algorithm can be selected and the advantages of all of them are effectively combined. This is the idea behind self-validation. For each case, first the self-validation process is performed on each algorithm on the candidate list. Performing well in the self-validation test indicates it is also very likely to do an excellent job in the real situation, so the best-performing algorithm could be selected to fulfil the real task. The performance thus depends on the reliability of the self-validation process, which is going to be introduced next.

A method called “double interpolation” [9] is applied to fulfill the self-validation task. The process flow is shown in Fig. 1. Suppose an interlaced sequence and a deinterlacing algorithm are given. Take an odd-lined field I_{odd} from the sequence for example. First, I_{odd} will be deinterlaced with the given algorithm to get a full-resolution frame D_{odd}. Then, D_{odd} will be interlaced and an even-lined field I_{even} will be derived. Next, the same deinterlacing algorithm is performed again on

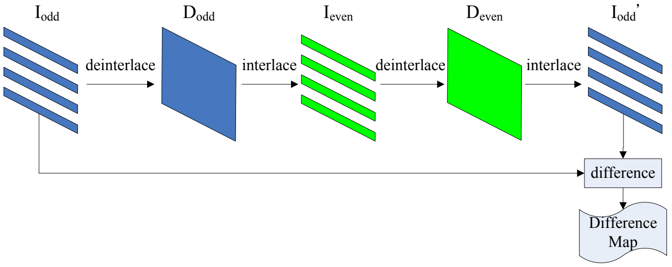


Fig. 1. Flow diagram of double interpolation.

I_{even} to get a full-resolution frame D_{even}. Finally, D_{even} will be interlaced to an odd-lined field I_{odd}'. Note that although this field is again odd-lined, it is not the original field anymore. Instead, it is obtained through the interpolation from the even-lined field that is derived from the original field. This new field I_{odd}' is compared with the original field I_{odd} to work out the differences between them for each pixel. The same process is applied for an even-lined field, and eventually all the fields in the sequence could be judged. By doing so, a double-interpolation difference map (DI-map) can be generated. The magnitude in the DI-map reflects whether this algorithm is self-consistent; a large magnitude points to the inconsistency of the algorithm, which means this algorithm is not suitable in this region, and vice versa.

III. Implementation

If now a great number of algorithms could be utilized, each algorithm could be examined with the self-validation process in section II. Therefore, the most suitable algorithm for each pixel could be decided independently, leading to an adaptive algorithm. In our real work, four algorithms were selected as the candidate methods: VTMF3 (three-tap vertical-temporal median filter) [10], VTMF7 (seven-tap vertical-temporal median filter) [11], INT, and ELA [12]. INT and ELA are exploited in a way that each direction of interpolation is considered as an input algorithm, rather than exploiting the result of only the minimum difference direction as in the original methods. The block diagram is shown in Fig. 2. These four algorithms are extremely fast so even that all of them need to be performed twice, the overall complexity is still very low. More algorithms could also be included and theoretically, the more input algorithms, the better the result should be.

For each input sequence, first one algorithm is applied to generate the deinterlaced result as a candidate. Then the self-validation process described in section II is performed to derive the DI-map for the selected algorithm. Next, for each pixel, all the DI-map values in a spatial as well as temporal window neighboring the current pixel are summed. This sum will indicate the “cost” for choosing this algorithm; the lower it is, the more likely it will be picked. In our experiment, the process for Y component and that for UV components are performed separately. That means the summation of Y-components in the DI-map determines the algorithm’s choice of Y, and the summation of U- and V- components in the DI-map determines the algorithm’s choice of both U and V, so their choices of algorithms can be different. The costs of U- and V-components are combined since in a 4:2:0 sequence they are more sparse than the Y-component, so combining them can reveal more information when making the decision. This was repeated for all the input algorithms. Finally, the result of the algorithm that generates the least difference sum will be chosen as the final result. In other words, for a pixel p, its best fitting algorithm’s index i for the Y-component and its Y-value Y_p are calculated as

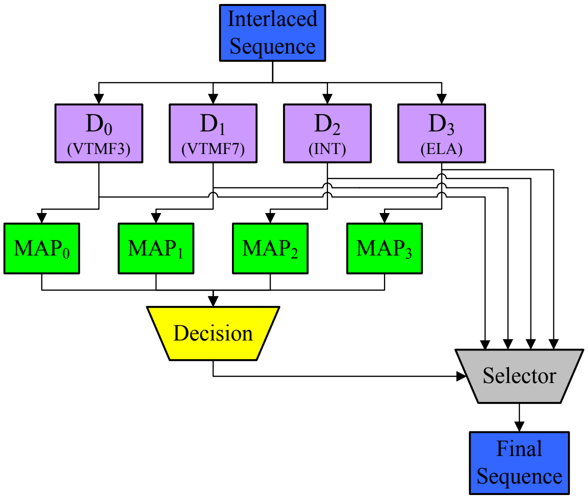


Fig. 2. Block diagram of the proposed self-validation framework.

$$i = \arg\min_{k} \sum_{q\in\Omega_Y} MY_{kq}^2, k = 0,1,2,3, \quad (1)$$

$$Y_p = Y_{ip}, \quad (2)$$

where MY_{kq} is the q-th value on the k-th Y-component DI-map, Ω_Y denotes the spatial and temporal window for Y-component, and Y_{ip} is the Y-component value of pixel p on the deinterlaced frame obtained by the i-th algorithm. On the other hand, the best fitting algorithm index j for the U- and V-components and its U- and V-values U_p and V_p are calculated as

$$j = \arg\min_{k} \sum_{q\in\Omega_{UV}} \left(MU_{kq}^2 + MV_{kq}^2\right), k = 0,1,2,3, \quad (3)$$

$$U_p = U_{jp}, \quad (4)$$

$$V_p = V_{jp}, \quad (5)$$

where MU_{kq} and MV_{kq} are the q-th values on the k-th U- and V-component DI-map respectively, Ω_{UV} denotes the spatial and temporal window for UV-components, and U_{jp} and V_{jp} are the U- and V-component values of pixel p on the deinterlaced frame obtained by the j-th algorithm, respectively.

IV. Results

The only parameter in the proposed framework is the window size in the summing process. For the spatial window size, the window size of Y component is set to 9×4, and the window size of the UV components is set to 21×10. The window size is not square since the input sequence is halfresolutioned, so the window height is only half the length of its width. The temporal window size is set to 3, which means for each frame, the previous and the following frames will also be taken into account. The window sizes here do not play an important role in a sense that the PSNRs of the results change only slightly when the window sizes are tuned in our experiment. In fact, the average PSNRs of the Y component only vary within 0.2 dB as the window size is changed from 5×2 to 25×12. Similar results can be obtained for the UV components. This is a desired result because it demonstrates that the proposed method is free of parameter constraint.

To evaluate the performance of the proposed algorithm, the deinterlaced results of 10 CIF sequences and the first 100 frames of 5 HD sequences are compared with those obtained by the input algorithms and some state-of-the-art algorithms. The PSNR values are shown in Tables I and II. It can be seen that the PSNRs of the proposed algorithm outperform the others by a substantial amount. Fig. 4 shows the deinterlaced result and the algorithm mode map for one frame in the "Table" sequence. It is clear that the proposed algorithm made the right choice when choosing between those algorithms, which demonstrates the validity of double-interpolation. Fig. 5-Fig. 6 show the comparison of some sample results of the proposed algorithm and other algorithms. In addition to better objective PSNRs, the subjective views also improve a lot.

Furthermore, a subjective test including 20 non-experts was performed to evaluate the visual quality. The MSU perceptual video quality tool [13] was used to test 3 CIF sequences and an HD sequence, and the test type of DSCQS (Double Stimulus Continuous Quality Scale) [14] was selected. In the test, the subject would see two sequences, the original and the deinterlaced one, but only one of them would be displayed at the same time. The subject could switch between them rapidly, and could watch them as many times as they wanted. After that, the subject would be asked to rate each video from score 1 to score 5, i.e., very annoying to indistinguishable. The subjects were not informed which the original sequence was, and the sequence order was random for each test, ruling out the possibility of direction-biased choices. The average scores and their standard deviations are shown in Fig. 3. To verify the reliability, a one-way ANOVA (Analysis of Variance) [15] was conducted. For all four sequences, the proposed algorithm turned out to show significant differences than the others ($p<0.01$) by the Tukey HSD test. Besides, most of the subjects gave score 5 to the results of the proposed algorithm, meaning that they are indistinguishable from the original sequences to them. Note that in the test the two videos could be switched rapidly instead of displayed at different positions or one after the other, so any subtle differences could be seen extremely easily. Besides, some subjects even gave the original sequence a lower score when compared with the result generated by the proposed method, and if we average the scores when these two are compared, the "Hall monitor" sequence obtained by the proposed algorithm would receive a higher score (4.95 to 4.80), while the deinterlaced result of "Parkrun" would get the same score as the original one (both 4.85).

TABLE I
AVERAGE PSNR (DB) OF DIFFERENT METHODS FOR 10 CIF SEQUENCES

Sequence	VTMF7 [12]	INT	ELA [13]	CAVTF [7]	MAMNN [8]	Proposed
Akiyo	44.11	45.92	38.33	49.57	48.07	49.16
Coastguard	30.55	31.57	28.50	33.01	34.54	37.34
Container	33.92	41.19	28.61	45.08	43.87	47.43
Flower	23.75	23.53	22.23	26.23	31.70	33.16
Foreman	34.10	32.30	30.33	36.08	35.77	37.85
Hall Monitor	37.38	38.52	31.64	42.86	41.01	42.06
Mother	43.53	41.91	38.42	47.39	46.65	47.41
Silent	38.78	36.48	32.30	43.19	41.20	43.87
Stefan	27.26	22.96	27.28	28.66	27.24	28.56
Table	31.33	30.71	28.10	37.59	35.44	37.78
Average	34.47	34.51	30.57	38.97	38.55	**40.46**

TABLE II
AVERAGE PSNR (DB) OF DIFFERENT METHODS FOR 5 HD SEQUENCES

Sequence	VTMF7 [12]	INT	ELA [13]	MA [4]	Proposed
Parkrun	25.40	21.56	25.79	25.74	30.49
Shields	33.29	26.64	33.22	32.90	38.94
Stockholm	33.52	28.20	32.75	33.19	37.70
Bluesky	35.46	24.26	38.29	38.42	41.35
Sunflower	44.00	31.40	43.78	41.57	46.50
Average	34.33	26.41	34.77	34.36	**39.00**

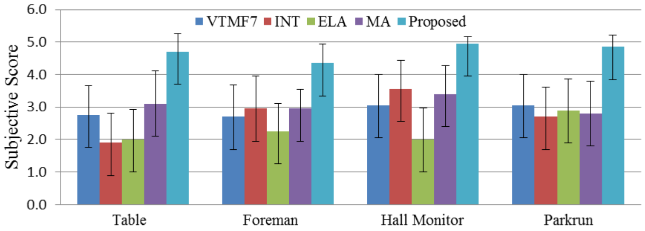
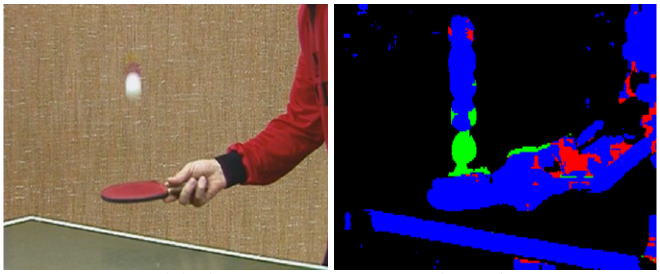


Fig. 3. The average subjective scores of different algorithms for selected sequences.

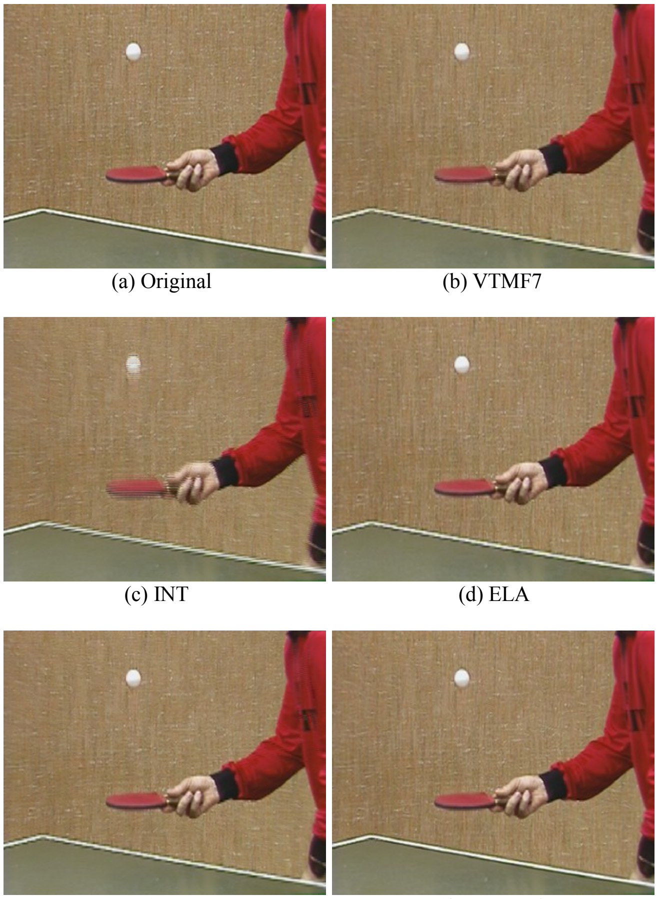
V. CONCLUSION

In this paper, a self-validation framework is proposed to solve the ill-posed deinterlacing problem. Previous works tend to make some assumptions about the original sequences, so their performances depend on how well the input sequences satisfy those requirements. The proposed approach, on the other hand, can make use of different deinterlacing algorithms by evaluating their performances at each pixel using a process called double interpolation. The final pixel value would be that generated by the algorithm demonstrating highest local consistency. By doing so, once the input sequence satisfies the hypotheses of one of the input algorithms, the choosing process should be able to pick it out and generate excellent result. The only parameter in this proposed framework is the size of the decision window, which does not play an important role on the output PSNRs, so the proposed algorithm is free of parameter constraints. The experimental results show that the proposed approach outperforms the other methods in both average objective quality measures and subjective visual quality tests.



(a) Original (b) Mode map

Fig. 4. (a) The original frame. (b) The corresponding algorithm decision mode map for the proposed framework. The red part represents VTMF3, green part represents VTMF7, blue part represents all directions of ELA, and black part represents all directions of INT. It can be seen that the proposed algorithm made the right choice when choosing between them, interpolating the static region using INT while utilizing ELA or other algorithms in dynamic areas.

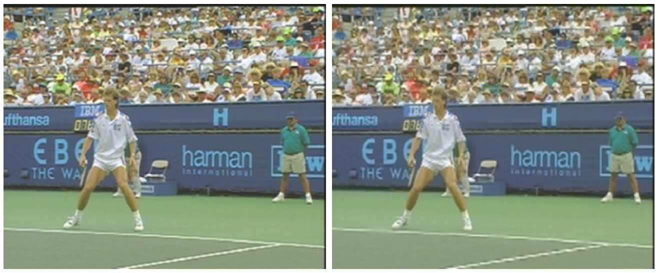


(a) Original (b) VTMF7

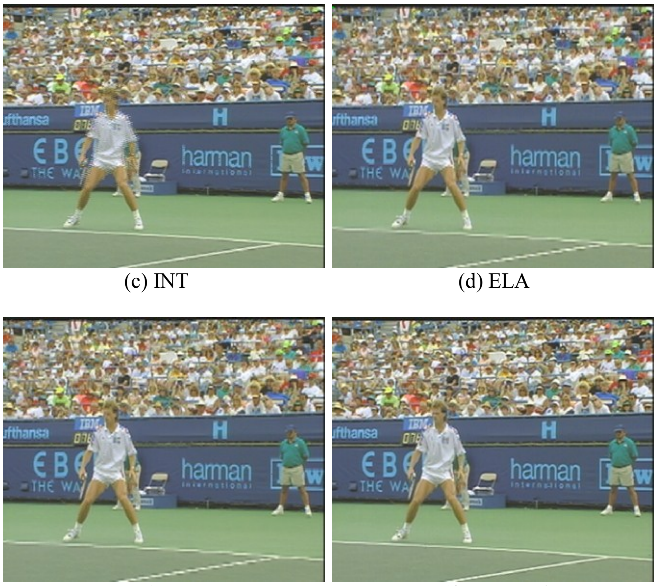
(c) INT (d) ELA

(e) MA (f) Proposed

Fig. 5. Results of different deinterlacing methods for sequence "Table."



(a) Original (b) VTMF7



(c) INT (d) ELA

(e) MA (f) Proposed

Fig. 6. Results of different deinterlacing methods for sequence "Stefan."

REFERENCES

[1] E. B. Bellers and G. de Haan, *De-interlacing—A key technology for scan rate conversion*. Amsterdam, The Netherlands: Elsevier, 2000.

[2] C. J. Kuo, C. Liao, and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 317–321, Apr. 1996.

[3] O. Kalevo and P. Haavisto, "Motion compensated deinterlacing," in *Proc. 1993 IEEE Int. Conf. Consumer Electron.*, pp. 40–41, Jun. 1993.

[4] S.-F. Lin, Y.-L. Chang and L.-G. Chen, "Motion adaptive interpolation with horizontal motion detection for deinterlacing," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, Nov. 2003.

[5] Y.-C. Fan, H.-S. Lin, A. Chiang, H.-W. Tsao, and C.-C. Kuo, "Motion compensated deinterlacing with efficient artefact detection for digital television displays," *Journal of Display Technology*, vol. 4, no. 2, Jun. 2008.

[6] S. Keller, F. Lauze, and M. Nielsen, "Deinterlacing using variational methods," *IEEE Trans. Image Processing*, vol. 17, no. 11, pp. 2015-2028, Nov. 2008.

[7] K. Lee and C. Lee, "High quality deinterlacing using content adaptive vertical temporal filtering," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, Nov. 2010.

[8] H. Choi and C. Lee, "Motion adaptive deinterlacing with modular neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 6, Jun. 2011.

[9] Y.-N. Liu, T.-C. Lin, and S.-Y. Chien, "A no reference quality evaluation method for CFA demosaicking," in *Digest of Technical Papers of International Conference on Consumer Electronics*, pp. 365–366, Jan. 2010.

[10] M. J. J. C. Annegarn, T. Doyle, P. H. Frencken and D. A. van Hees, "Video signal processing circuit for processing an interlaced video signal," U.S. Patent 4 740 842, Apr. 26, 1988.

[11] J. Salo, Y. Nuevo and V. Hameenaho, "Improving TV picture quality with linear-median type operations", *IEEE Transactions on Consumer Electronics*, vol. 34, no 3, pp. 373-379, Aug. 1988.

[12] T. Doyle and M. Looymans, "Progressive scan conversion using edge information," in *Signal Processing of HDTV II,* L. Chiariglione, Ed. Amsterdam, The Netherlands: Elsevier, 1990, pp. 711–721.

[13] D. Vatolin and M. Smirnov. (2011) MSU perceptual video quality tool [Online].Available:http://compression.ru/video/quality_measure/perceptual_video_quality_tool_en.html

[14] ITU-R Recommendation BT.500-11, "Methodology for the subjective assessment of the quality of television pictures," 2002.

[15] R. Lowry. (2012) One-way analysis of variance. [Online]. Available: http://faculty.vassar.edu/lowry//anova1u.html