J. Vis. Commun. Image R. 25 (2014) 466–477

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Quality-efficient syntax element-based de-interlacing method for H.264-coded video sequences with various resolutions

Wei-Jen Yang [a], Kuo-Liang Chung [a,1], Yong-Huai Huang [b,*,2], Le-Chung Lin [a]

[a] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC
[b] Institute of Computer and Communication Engineering, Department of Electronic Engineering, Jinwen University of Science and Technology, No. 99, An-Chung Road, Hsin-Tien Dist., New Taipei City 23154, Taiwan, ROC

ARTICLE INFO

Article history:
Received 20 November 2012
Accepted 10 December 2013
Available online 26 December 2013

Keywords:
De-interlacing
Field image
H.264/AVC
Inter mode
Motion vector
Quality
Syntax elements
Various resolutions

ABSTRACT

In this paper, we propose a novel quality-efficient de-interlacing method for H.264-coded video sequences with various resolutions. In the proposed method, using the syntax elements provided in H.264 bitstreams, four new and efficient strategies are delivered for inter-coded macroblocks to improve the quality of de-interlaced video sequences as well as alleviate the error propagation side effect. Based on the real and generated interlaced video sequences with various common resolutions, experimental results demonstrate the proposed de-interlacing method achieves better quality in terms of both objective and subjective measures when compared with the recently published method by Dong and Ngan.

1. Introduction

De-interlacing technique is used to convert the fields of an interlaced video sequence into the frames of a de-interlaced video sequence [10,16]. Besides filling missing lines in the fields, successful de-interlacing methods also aim to achieve quality benefits.

Over the past two decades, a number of de-interlacing methods were developed. These developed methods can be classified into three categories, the intra-field methods [8,10,12,13,17], the inter-field methods [5,6,11,14], and the hybrid methods [2–4,9,15]. In the intra-field methods, only the spatial information in the current field is exploited to fill missing pixel values. This kind of de-interlacing methods has low computation benefit, but it often produces unsatisfactory de-interlaced results, especially in the edges or motion regions. The inter-field methods utilize the temporal motion information between the adjacent fields to fill the missing pixel values. They often can yield good quality of de-interlaced results when the temporal motion information is reliable; however, unreliable motion information may result in undesirable de-interlaced results and high computational complexity is required in the motion estimation process. The hybrid methods take both spatial and motion information into account. They often yield better quality performance than intra-field and inter-field methods, but they sometimes suffer from high computational load.

Nowadays, since H.264/AVC [1] has already been a popular video compression standard, developing an efficient de-interlacing method specifically for H.264-coded interlaced video sequences is an interesting and important issue. To de-interlace an H.264-coded interlaced video sequence, we can first decode the H.264 bitstream to obtain a reconstructed interlaced video sequence, and then apply existing de-interlacing methods to fill the missing lines of each field. However, the syntax elements (SEs), such as motion vectors (MVs) and the residual of each block, which include important hints of motions and edges, cannot be used in the de-interlacing process. Since the SEs gives many useful information for de-interlacing, for H.264-coded interlaced video sequences, Dong and Ngan [7] proposed the first de-interlacing method by efficiently utilizing the SEs in the H.264 bitstreams. Their proposed quality-efficient de-interlacing method confines to full high-definition (HD) resolution. Unfortunately, the quality degradation problem may happen when directly applying their method to H.264-coded video sequences with lower resolutions, such as the commonly used common international format (CIF), quarter CIF (QCIF), standard-definition (SD), and so forth. In real applications,

* Corresponding author.
E-mail addresses: klchung01@gmail.com (K.-L. Chung), yonghuai@ms28.hinet.net (Y.-H. Huang).

[1] Supported by National Council of Science of R.O.C. under contracts NSC 99-2221-E-011-078-MY3 and NSC 101-2221-E-011-139-MY3.
[2] Supported by the National Science Council of R.O.C. under the contracts NSC100-2221-E-228-007 and NSC101-2221-E-228-010.

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.12.014

for example, for video sequences with format 576i, each field is originally interlaced and it must be de-interlaced in advance before playing. As a result, de-interlacing for H.264-coded interlaced video sequences with various resolutions is necessary, leading to the main motivation of this research.

This paper presents a novel SE-based de-interlacing method for H.264-coded interlaced video sequences with various resolutions. In the proposed method, to fully utilize the SEs, such as MVs, residual, discrete cosine transform (DCT) coefficients, and indices of the reference frames for inter-coded macroblocks (MBs), four new de-interlacing strategies, namely (1) the quarter-pixel precision-based motion and residual compensations, (2) the adaptive selection of reference frame to reduce error propagation, (3) the motion blur propagation reduction for full HD video sequences, and (4) the motion compensation- and low-pass filter-based quality refinement, are presented to improve the de-interlaced video quality as well as alleviate the error propagation side effect in the video sequences. We perform the experiment on the real interlaced video sequences and the interlaced video sequences generated by removing the lines of progressive videos. Experimental results demonstrate the proposed de-interlacing method delivers better quality of de-interlaced video sequences than Dong and Ngan's method. Based on the generated interlaced video sequences, the average peak signal-to-noise ratio (PSNR) quality improvement of the proposed de-interlacing method over Dong and Ngan's method for the video sequences with the resolutions of QCIF, CIF, SD, and full HD can achieve, respectively, more than 1.86, 1.33, 0.59, and 0.88 dBs. Based on the real interlaced video sequences, the proposed method yields fewer aliasing artifacts than Dong and Ngan's method, leading to better visual quality of de-interlaced video sequences.

The rest of this paper is organized as follows. In Section 2, the proposed de-interlacing method consisting of four efficient strategies is presented. In Section 3, we report the experimental results on the quality superiority of the proposed method for the video sequences with various resolutions. Section 4 addresses some concluding remarks.

2. The proposed SE-based de-interlacing method for H.264-coded video sequences with various resolutions

For the k-th $M \times N$ H.264/AVC decoded field f^k, depicted in Fig. 1(a), denote by $f^k(i,j)$ the value of the pixel at position (i,j), where i denotes the vertical axis and j denotes the horizontal axis. Initially, enlarge f^k, by zero-padding approach, to obtain a $2M \times N$ frame F^k, depicted in Fig. 1(b). It yields to

$$F^k(i,j) = \begin{cases} f^k(\frac{i}{2},j), & \text{if } i \text{ and } k \text{ are even,} \\ f^k(\frac{i-1}{2},j), & \text{if } i \text{ and } k \text{ are odd,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Since a MB is of size 16×16 in H.264/AVC, a field is thus partitioned into a set of MBs. Denote by b_m^k the m-th MB in f^k and the corresponding enlarged MB of size 32×16 in F^k is denoted by B_m^k. For easy understanding, B_m^k can be viewed as a working domain.

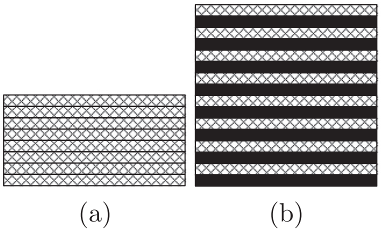


Fig. 1. Depiction of (a) f^k and (b) F^k.

According to the rate-distortion (RD) cost criterion, a MB b_m^k is encoded by either the intra or inter mode. When b_m^k in f^k is encoded by the intra mode, we apply the direction-based median filters in Dong and Ngan's method [7] to de-interlace zero-padded pixels although it can be extended to further consider two extra edge directions with −22.5 degree and 22.5 degree in the case of horizontal prediction mode. When b_m^k in f^k is encoded by the inter mode, the proposed four new de-interlacing strategies, which consist of (1) the quarter-pixel precision-based motion and residual compensations, (2) the adaptive selection of reference frame to reduce error propagation, (3) the motion blur propagation reduction for full HD video sequences, and (4) the motion compensation- and low-pass filter-based quality refinement, are exploited to improve the quality of de-interlaced video sequences as well as alleviate the error propagation side effect. In what follows, we first present the proposed four new strategies one by one, and then provide the whole de-interlacing process for inter-coded blocks. At the end of this section, the computational time comparison between Dong and Ngan's and the proposed methods is provided. It is shown that the computational time performance of proposed method is competitive to Dong and Ngan's method.

2.1. Quarter-pixel precision-based motion and residual compensations

In H.264/AVC, an inter-coded MB b_m^k in f^k is partitioned into a set of blocks in which each block may be of size 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, or 4×4. For easy exposition, if there exists any non-4×4 block in b_m^k, we further partition it into 4×4 blocks and assign its MV to the associated 4×4 blocks. Denote by $b_{m,n}^k$, where $0 \leqslant n \le 15$, the n-th partitioned 4×4 block of b_m^k, and the corresponding enlarged 8×4 block in F^k is denoted by $B_{m,n}^k$.

According to the temporal correlation in a video sequence, Dong and Ngan de-interlace $B_{m,n}^k$ by using the motion compensation, which considers the MVs with integer-pixel precision and utilizes the nearest de-interlaced frame as the reference frame. Because H.264 does support the MV with quarter-pixel precision and five reference frames; the decoder can directly obtain the related information from the SEs. We thus consider the MV with quarter-pixel precision and five reference frames instead of the MV with integer-pixel precision and one reference frame, leading to better motion compensation effect; in addition, the residual compensation is exploited to further improve the quality of the de-interlaced video sequence.

Since each MV in H.264 is represented by quarter-pixel precision, the actual horizontal and vertical displacements of a decoded MV, $(\Delta i, \Delta j)$, are $\frac{\Delta i}{4}$ and $\frac{\Delta j}{4}$, respectively. For easy explanation, we utilize actual displacements to represent the MV of $B_{m,n}^k$. As mentioned before, the width of F^k is the same as that of f^k, but the height of F^k is twice as long as that of f^k. Therefore, the MV of $B_{m,n}^k$ can be approximated by inheriting the horizontal displacement of $b_{m,n}^k$'s MV and scaling the vertical displacement of $b_{m,n}^k$'s MV. Denoting, respectively, by $(\Delta i_{m,n}^k, \Delta j_{m,n}^k)$ and f^r the decoded MV and the reference field of $b_{m,n}^k$, the MV of $B_{m,n}^k$ can be calculated by

$$(\Delta \hat{i}_{m,n}^k, \Delta \hat{j}_{m,n}^k) = \left(\text{round}\left(\frac{S(\Delta i_{m,n}^k)}{4} \right), \frac{\Delta j_{m,n}^k}{4} \right), \tag{2}$$

where

$$S(\Delta i_{m,n}^k) = \begin{cases} 2(\Delta i_{m,n}^k - 2), & \text{if } f^k \text{ is a bottom field and } f^r \text{ is a top field,} \\ 2(\Delta i_{m,n}^k + 2), & \text{if } f^k \text{ is a top field and } f^r \text{ is a bottom field,} \\ 2\Delta i_{m,n}^k, & \text{otherwise.} \end{cases} \tag{3}$$

Note that the model for scaling the vertical displacement used in the above formulas follows that by Dong and Ngan [7]. However, as for the vertical displacement, since the vertical resolution of an interlaced field is only half of the original frame, it is ineffectual when using quarter-pixel precision for de-interlacing. Therefore, we use quarter-pixel precision for the horizontal displacement, but integer-pixel precision for the vertical displacement.

Based on the MV of $B^k_{m,n}$ and the de-interlaced reference frame F^r, the zero-padded pixels in $B^k_{m,n}$ are replaced by those pixels using the motion compensation. Besides that, the decoded residual of the field f^k, say r^k, which is not considered in Dong and Ngan's method, can be used to further improve the de-interlaced video quality by the residual compensation. Since the size of r^k is the same as that of f^k, to exploit r^k to compensate the de-interlaced result, we should enlarge vertical resolution of r^k to obtain the enlarged residual R^k. Generally, since r^k is quite homogeneous, we first enlarge r^k, by zero-padding approach, to obtain R^k, and then interpolate the zero-padded pixel by using the following low-pass filter [7]:

$$R^k(i,j) = \frac{1}{8}\left[5\sum_{a\in\{\pm1\}} R^k(i+a,j) - \sum_{b\in\{\pm3\}} R^k(i+b,j)\right]. \tag{4}$$

According to the above description, the zero-padded pixels in $B^k_{m,n}$ can be de-interlaced by

$$\dot{F}^k(i,j) = F^r(i+\Delta\hat{i}^k_{m,n}, j+\Delta\hat{j}^k_{m,n}) + R^k(i,j). \tag{5}$$

From Table 2 in Section 3, the video sequences de-interlaced by the proposed quarter-pixel precision-based motion and residual compensations have 1.63-5.44 dBs PSNR improvement when compared with the video sequences de-interlaced by the motion compensation with the integer-pixel precision MV but without residual compensation.

2.2. Adaptive selection of reference frame to reduce error propagation

The motion compensation-based de-interlacing strategy mentioned above may cause error propagation side effect, which leads to the degradation of the de-interlaced video quality. Thus, we propose a new strategy to select the reference frame adaptively to reduce the error propagation when the current field and the reference field are both top fields or bottom fields. We take Fig. 2 to explain why such a kind of errors are propagated and how to deal with this kind of error propagation. As shown in Fig. 2, considering three successive frames, F^{k-2}, F^{k-1}, and F^k, the H.264-decoded pixels, the de-interlaced pixels, and the untreated pixels are, respectively, marked by the symbols "○," "△," and "×".

Suppose the current block $B^k_{m,n}$ in F^k has the best matched block with MV, $(4,-2)$, in the reference frame F^{k-2}. The untreated pixels in $B^k_{m,n}$ can be de-interlaced by referring the de-interlaced pixels in the reference block. Unfortunately, the error caused by de-interlacing process in the reference block will be propagated to $B^k_{m,n}$, resulting in the error propagation problem. To reduce this kind of error propagation, we take F^{k-1} as the reference frame and modify the original MV, $(4,-2)$, to the new MV, $(2,-1)$, depicted by blue arrow in Fig. 2, directly based on the linear motion assumption among successive frames [9]. Thus, the reference block with MV, $(2,-1)$, in F^{k-1} is used to de-interlace $B^k_{m,n}$. Since the values of the untreated pixels in $B^k_{m,n}$ are inherited from those of the H.264-decoded pixels rather than the de-interlaced pixels, the error propagation side effect can be reduced.

From Fig. 2, it is observed that after de-interlacing $B^k_{m,n}$, the error propagation may happen when the vertical displacement of the MV, $\Delta\hat{i}^k_{m,n}$, is even. Therefore, the adaptive selection of the reference frame and the corresponding MV modification are realized by selecting the reference frame F^{r+1} instead of F^r and modifying the MV by

$$(\Delta\bar{i}^k_{m,n}, \Delta\bar{j}^k_{m,n}) = \frac{k-r-1}{k-r}\left(\Delta\hat{i}^k_{m,n}, \Delta\hat{j}^k_{m,n}\right), \tag{6}$$

respectively, when $\frac{k-r-1}{k-r}\Delta\hat{i}^k_{m,n}$ is even. When $\frac{k-r-1}{k-r}\Delta\hat{i}^k_{m,n}$ is odd, we still utilize the originally selected reference frame and the related MV to de-interlace $B^k_{m,n}$. According to our experiments, the influence of the proposed adaptive reference frame selection and MV modification strategy on the de-interlaced video quality decreases gradually with the increase of the temporal distance between F^k and F^r, implying that the optimal quality improvement introduced by this strategy often occurs when $F^r = F^{k-2}$. Thus, we only enable the reference frame selection and MV modification strategy when $F^r = F^{k-2}$. According to experimental results in Section 3, selecting the proper reference frame can further improve PSNR by 0.26-5.09 dBs.

2.3. Motion blur propagation reduction for full HD video sequences

It is known that the motion blur effect on de-interlaced full HD video sequences is more serious than that on de-interlaced video sequences with lower resolutions. Motion blur propagation often occurs when using the blurred reference block to de-interlace $B^k_{m,n}$. Since the blocks in a full HD video sequence usually tend to be smooth, de-interlacing the high motion blocks by using the low-pass filter in Eq. (4) can achieve satisfactory de-interlaced video quality as well as does not introduce the motion blur propagation. Therefore, when $|\frac{\Delta\hat{i}^k_{m,n}}{k-r}| + |\frac{\Delta\hat{j}^k_{m,n}}{k-r}| > 15$, the block $B^k_{m,n}$ is treated as a high motion block and it is de-interlaced by the low-pass filter instead of using the motion compensation approach.

Applying the proposed motion blur propagation reduction strategy to the full HD video sequences, experimental results shown in Table 2 of Section 3 demonstrate that it further improves PSNR by about 0.1 dBs in average.

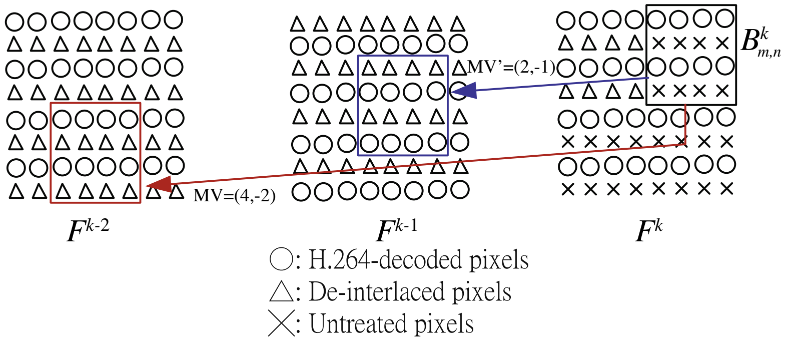


Fig. 2. Depiction of error propagation and MV modification.

2.4. Motion compensation- and low-pass filter-based quality refinement

According to the motion compensation-based de-interlacing formula in Eq. (5), it is observed that the quality of a de-interlaced video sequence is affected by the MV. A defective MV may result in the degradation of the de-interlaced video quality, and thus it is necessary to verify the reliability of the MV. When the MV is unreliable, we should refine the de-interlaced result to further improve the de-interlaced video quality.

In the motion compensation-based de-interlacing, a reliable MV often results in a small sum of absolute values of the corresponding decoded residual. Denoting by $r^k_{m,n}$ the corresponding 4×4 residual block of $b^k_{m,n}$, the sum of absolute values of the decoded residual can be expressed as

$$\mathrm{SAR}(r^k_{m,n}) = \sum_{i=0}^{3}\sum_{j=0}^{3}|r^k_{m,n}(i,j)|. \tag{7}$$

If $\mathrm{SAR}(r^k_{m,n}) < T$, where T is set to 32 empirically, it indicates the MV is reliable, implying that the quality of the de-interlaced result by Eq. (5) is good enough for de-interlacing $B^k_{m,n}$; otherwise, the motion compensation- and low-pass filter-based quality refinement strategy is applied to refine the quality of the de-interlaced result.

In the proposed motion compensation- and low-pass filter-based quality refinement strategy, the refined de-interlaced pixel at location (i,j) in $B^k_{m,n}$ is expressed by

$$\breve{F}^k(i,j) = w\widehat{F}^k(i,j) + (1-w)\dot{F}^k(i,j), \tag{8}$$

where $\widehat{F}^k(i,j)$ and $\dot{F}^k(i,j)$ denote, respectively, the de-interlaced pixel using the low-pass filter by Eq. (4) and the motion compensation by Eq. (5). In what follows, we describe how to utilize the texture information of the residual, $r^k_{m,n}$, to determine the appropriate weight w in Eq. (8).

Referring to Fig. 3, it is clear that for a block with larger SAR, the block often has more fruitful texture information, which can be represented by the energy of the DCT coefficient matrix of $r^k_{m,n}$. The DCT coefficient matrix of the residual is denoted by $\mathcal{D}^k_{m,n}$, which is an SE provided in H.264 bitstream and can be directly obtained from the decoder side, and it also can be utilized to measure the texture information of $b^k_{m,n}$. We first partition $\mathcal{D}^k_{m,n}$ into four parts, namely the low frequency part, the vertical part, the horizontal part, and the high frequency part; as shown in Fig. 4, they are, respectively, represented by $\mathcal{L}^k_{m,n} = \{\mathcal{D}^k_{m,n}(u,v)|0 \leqslant u, v \leq 1\}$, $\mathcal{V}^k_{m,n} = \{\mathcal{D}^k_{m,n}(u,v)|0 \leqslant u \leq 1, 2 \leqslant v \leq 3\}$, $\mathcal{H}^k_{m,n} = \{\mathcal{D}^k_{m,n}(u,v)|2 \leqslant u \leq 3, 0 \leqslant v \leq 1\}$, and $\mathcal{G}^k_{m,n} = \{\mathcal{D}^k_{m,n}(u,v)|2 \leqslant u, v \leq 3\}$. Then, the texture information of $b^k_{m,n}$ can be expressed by

$$EG(\mathcal{P}^k_{m,n}) = \sum_{\mathcal{D}^k_{m,n}(u,v)\in\mathcal{P}^k_{m,n}} |\mathcal{D}^k_{m,n}(u,v)|, \tag{9}$$

where $\mathcal{P}^k_{m,n}$ could be $\mathcal{L}^k_{m,n}$, $\mathcal{V}^k_{m,n}$, $\mathcal{H}^k_{m,n}$, or $\mathcal{G}^k_{m,n}$.

If the energy of high frequency part is small, it reveals that the texture of $b^k_{m,n}$ is not complex, indicating that the low-pass filter-based de-interlaced result by Eq. (4) and the motion compensation-based de-interlaced result by Eq. (5) can be used to refine the de-interlaced quality of $B^k_{m,n}$. Empirically, when $\frac{EG(\mathcal{G}^k_{m,n})}{\sum_{\mathcal{A}\in\{\mathcal{L}^k_{m,n},\mathcal{V}^k_{m,n},\mathcal{H}^k_{m,n},\mathcal{G}^k_{m,n}\}} EG(\mathcal{A})} < 0.1$, the de-interlaced pixels in $B^k_{m,n}$ are refined by the weighted average of results by Eqs. (4) and (5). Referring to Eq. (8), the associated weight could be determined by

$$w = \frac{EG(\mathcal{L}^k_{m,n}) + EG(\mathcal{V}^k_{m,n})}{\sum_{\mathcal{A}\in\{\mathcal{L}^k_{m,n},\mathcal{V}^k_{m,n},\mathcal{H}^k_{m,n},\mathcal{G}^k_{m,n}\}} EG(\mathcal{A})}. \tag{10}$$

The rationale behind the weight w is that since the low-pass filter in Eq. (4) is a vertical low-pass filter, it is suitable for de-interlacing the pixels in a homogeneous block or a vertical texture block, implying that the de-interlaced result by Eq. (4) should own more weight in refining the de-interlaced pixels for the block with large $EG(\mathcal{L}^k_{m,n})$ or $EG(\mathcal{V}^k_{m,n})$. However, when $\frac{EG(\mathcal{G}^k_{m,n})}{\sum_{\mathcal{A}\in\{\mathcal{L}^k_{m,n},\mathcal{V}^k_{m,n},\mathcal{H}^k_{m,n},\mathcal{G}^k_{m,n}\}} EG(\mathcal{A})} > 0.1$, the low-pass filter is ineffectual to refine the de-interlaced pixels since the texture direction of $b^k_{m,n}$ is complex. Therefore, we directly adopt the de-interlaced result by Eq. (5) as the final result for this case.

In Table 2 illustrated in Section 3, experimental results show that the proposed motion compensation- and low-pass filter-based quality refinement strategy further has 0.08-0.56 dBs PSNR improvement.

2.5. The whole de-interlacing process for inter coded blocks

Based on the four proposed new strategies described above, we now present the whole de-interlacing process for inter-coded blocks. Note that the current field f^k, reference field f^r, decoded residual r^k, and DCT coefficient matrix $\mathcal{D}^k_{m,n}$, can be directly obtained by the SEs during the decoding process. Given a 8×4 enlarged block $B^k_{m,n}$ in F^k and the corresponding reference frame F^r, the de-interlacing process for inter-coded blocks involves the following eight steps:

Step 1: Calculate, by Eq. (2), the MV of $B^k_{m,n}$, $(\Delta\hat{i}^k_{m,n}, \Delta\hat{j}^k_{m,n})$.

Step 2: If the input video sequence is a full HD video sequence and $|\frac{\Delta\hat{i}^k_{m,n}}{k-r}| + |\frac{\Delta\hat{j}^k_{m,n}}{k-r}| > 15$, de-interlace, by the low-pass filter in Eq. (4), the zero-padded pixels in $B^k_{m,n}$ and go to Step 8; otherwise, go to Step 3.

Step 3: If f^k and f^r are both top fields or bottom fields, go to Step 4; otherwise, go to Step 5.

Step 4: If $F^r = F^{k-2}$ and $(\frac{1}{2}\Delta\hat{i}^k_{m,n} \bmod)\ 2 = 0$, use F^{k-1} as the reference frame and modify the MV by Eq. (6); then, go to Step 5. Otherwise, go to Step 5 directly.

Step 5: De-interlace, by Eq. (5), the zero-padded pixels in $B^k_{m,n}$.

Step 6: Calculate $\mathrm{SAR}(r^k_{m,n})$ by Eq. (7). If $\mathrm{SAR}(r^k_{m,n}) > 32$, calculate $EG(\mathcal{L}^k_{m,n}), EG(\mathcal{V}^k_{m,n})$, $EG(\mathcal{H}^k_{m,n})$, and $EG(\mathcal{G}^k_{m,n})$ by Eq. (9) and go to Step 7; otherwise, go to Step 8.

Step 7: If $\frac{EG(\mathcal{G}^k_{m,n})}{\sum_{\mathcal{A}\in\{\mathcal{L}^k_{m,n},\mathcal{V}^k_{m,n},\mathcal{H}^k_{m,n},\mathcal{G}^k_{m,n}\}} EG(\mathcal{A})} < 0.1$, refine, by Eq. (8), the de-interlaced pixels in $B^k_{m,n}$ and go to Step 8; otherwise, go to Step 8 directly.

Step 8: Output the de-interlaced result.

2.6. Computational analysis

In this subsection, the computational comparison between Dong and Ngan's method and the proposed method is provided. For de-interlacing intra coded blocks, both methods have the same computational complexity because they use the same direction-based median filters.

For de-interlacing each inter coded block, both methods have a common calculation job, calculating the sum of absolute values of the decoded residual (see Eq. (7)). According to the four texture parts in DCT coefficient matrix, the proposed method performs the concerned three operations, such as the quarter-pixel precision-based motion and residual compensations, the adaptive selection of reference frame to reduce error propagation, and low-pass filter for full HD video sequences; Dong and Ngan's method mainly involves the three operations, such as the motion compensation, the low-pass filter, and the hybrid one. Examining the concerned three operations in each method, the first operation used in the

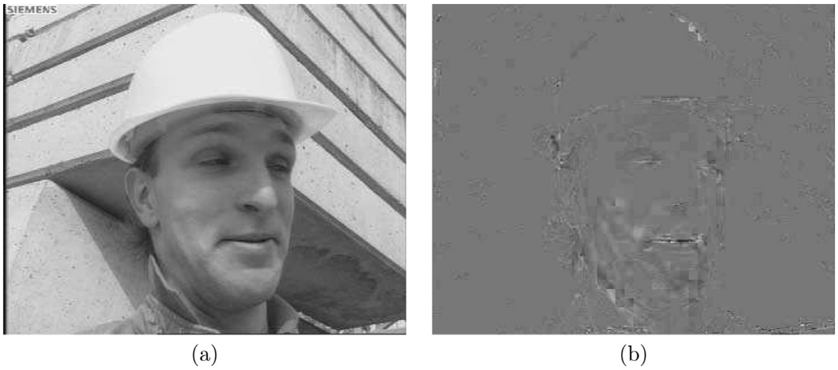

(a) (b)

Fig. 3. (a) Original and (b) decoded residual of Foreman frame.

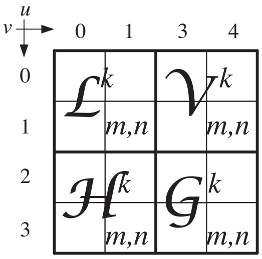


Fig. 4. Four texture parts, $\mathcal{L}^k_{m,n}$, $\mathcal{V}^k_{m,n}$, $\mathcal{H}^k_{m,n}$, and $\mathcal{G}^k_{m,n}$, in DCT coefficient matrix.

proposed method, which uses the low-pass filter to create sub-pixels and enlarge the vertical resolution of the decoded residual for motion and residual compensations, respectively, may result in more computational overhead than Dong and Ngan's method.

However, besides the computational time required in the de-interlacing process, we also should consider the computational time required in decoding the related SEs before performing the de-interlacing process, i.e., decoding the related inter-coded block. Basically, decoding each inter-coded block is rather time-consuming and mainly involves loading H.264 bitstream from storage media, decoding the entropy-coded data from the loaded bitstream, using the decoded MV to perform the motion compensation, decoding and de-quantizing the DCT coefficients of the residual block, and performing inverse DCT to reconstruct the

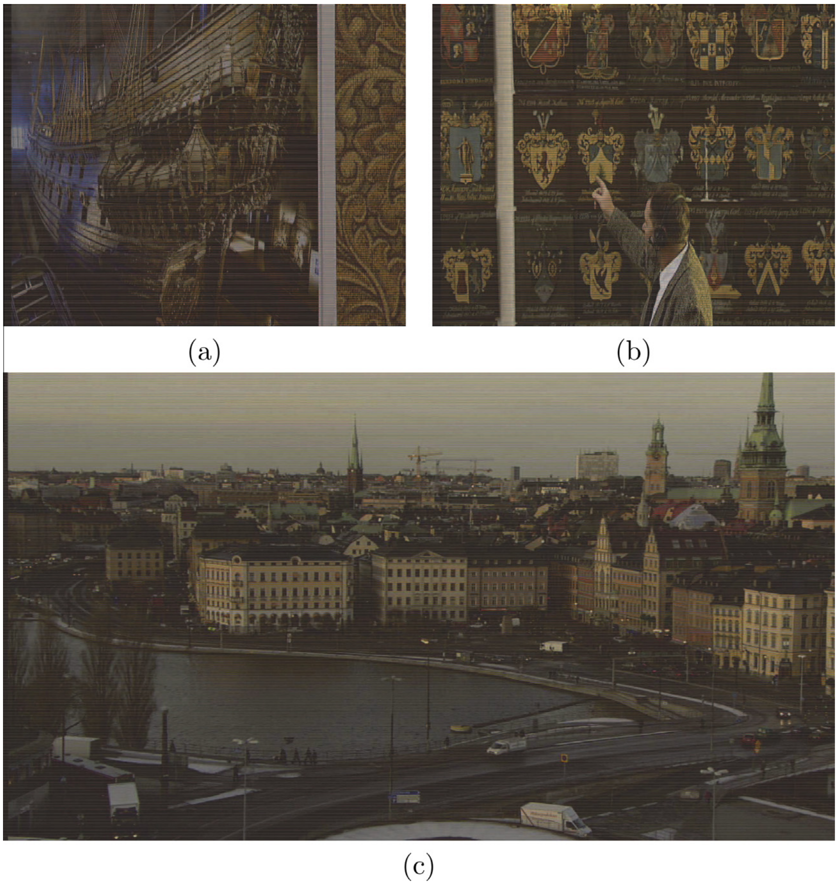
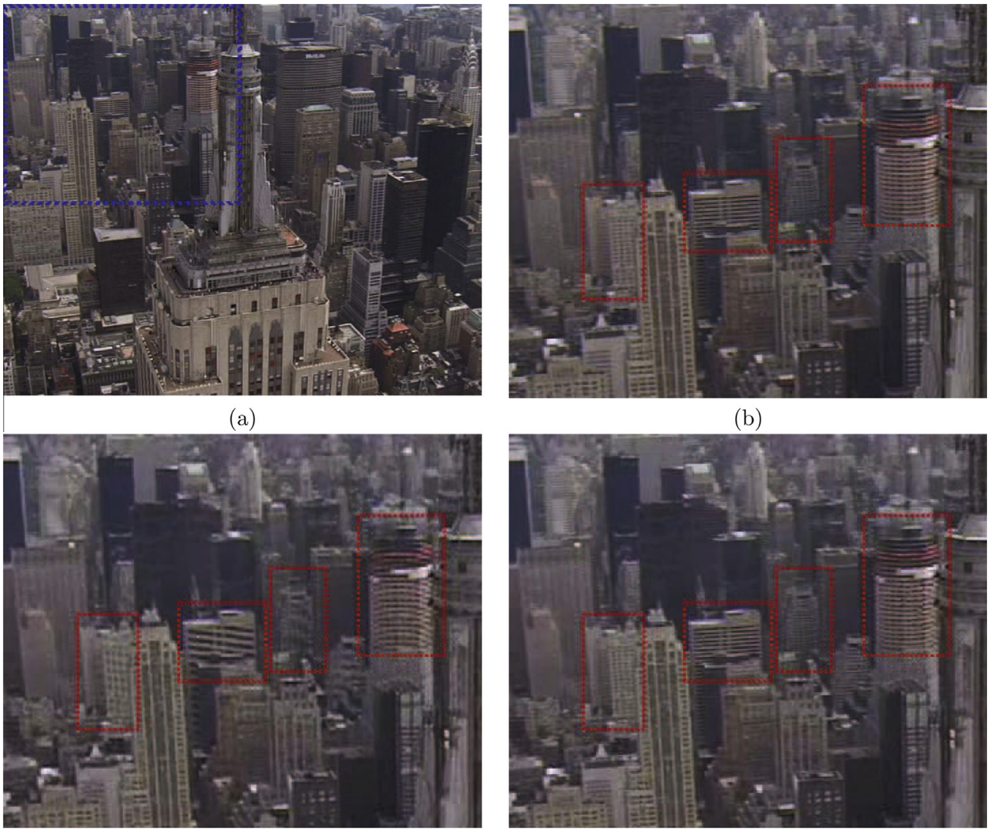

(a) (b) (c)

Fig. 5. Three test real interlaced videos: (a) Vasa ship calendar (576i), (b) Shields (576i), and (c) Stockholm (1080i).

Table 1
PSNRs (dB) of the de-interlaced video sequences using Dong and Ngan's method and the proposed method.

Resolution	Sequence	Dong and Ngan's method		The proposed method	
QCIF		128 kb/s	256 kb/s	128 kb/s	256 kb/s
	Akiyo	35.59	36.25	39.45	41.10
	Carphone	30.05	31.26	30.49	31.90
	Coastguard	25.10	25.90	27.88	28.85
	Foreman	29.89	31.19	30.55	32.29
	Mobile	21.72	22.57	23.29	24.83
	Average	28.47	29.44	30.33	31.80
CIF		768 kb/s	1 Mb/s	768 kb/s	1 Mb/s
	Bus	26.63	27.10	26.54	27.16
	Coastguard	27.55	28.11	28.63	29.12
	Flower	21.68	22.05	24.23	24.52
	Football	31.57	32.59	31.51	32.62
	Foreman	29.87	30.05	31.00	31.18
	News	34.95	35.73	38.34	38.98
	Average	28.71	29.27	30.04	30.60
SD		4 Mb/s			
	City	30.74		31.12	
	Ice	38.03		38.91	
	Soccer	33.51		34.00	
	Average	34.09		34.68	
full HD		20 Mb/s			
	Pedestrian area	39.97		40.45	
	Rush hour	40.65		41.47	
	Tractor	36.60		37.29	
	River bed	33.41		34.95	
	Average	37.66		38.54	



(a)

(b)

(c)

(d)

Fig. 6. (a) The original 391st frame of City video sequence; the magnified subframes of (b) the original frame and the corresponding subframes generated by (c) Dong and Ngan's method and (d) the proposed method.

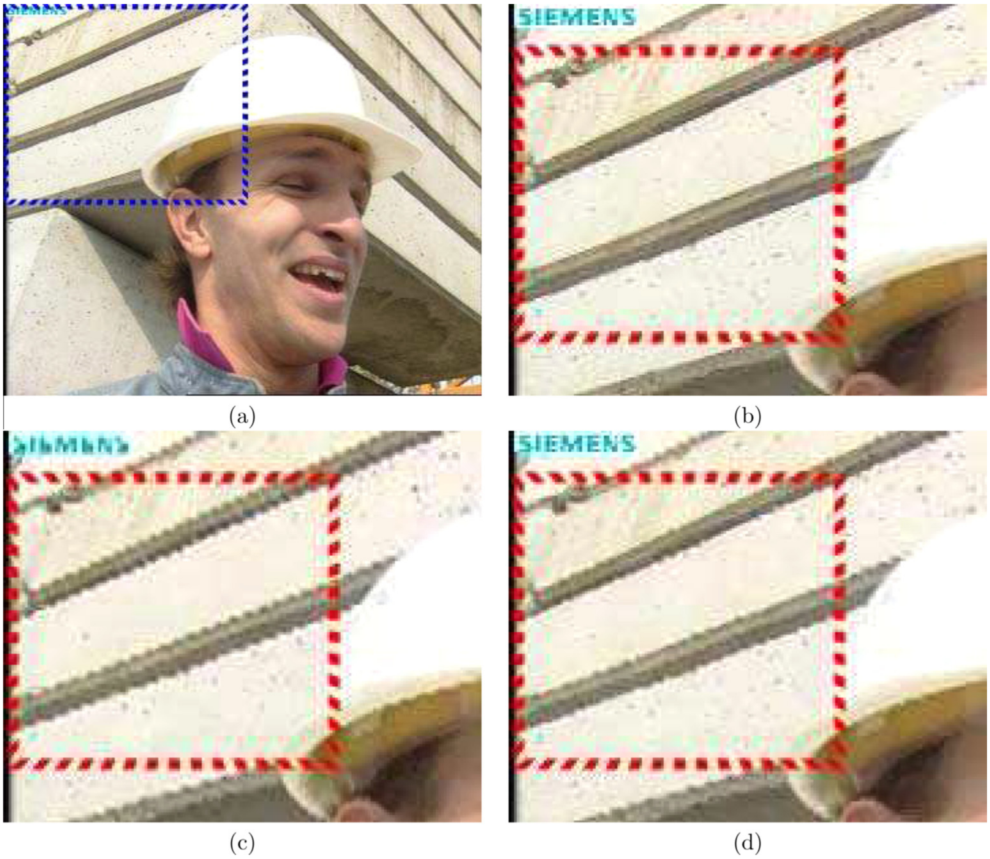


Fig. 7. (a) The original 12th frame of Foreman video sequence; the magnified subframes of (b) the original frame and the corresponding subframes generated by (c) Dong and Ngan's method and (d) the proposed method.

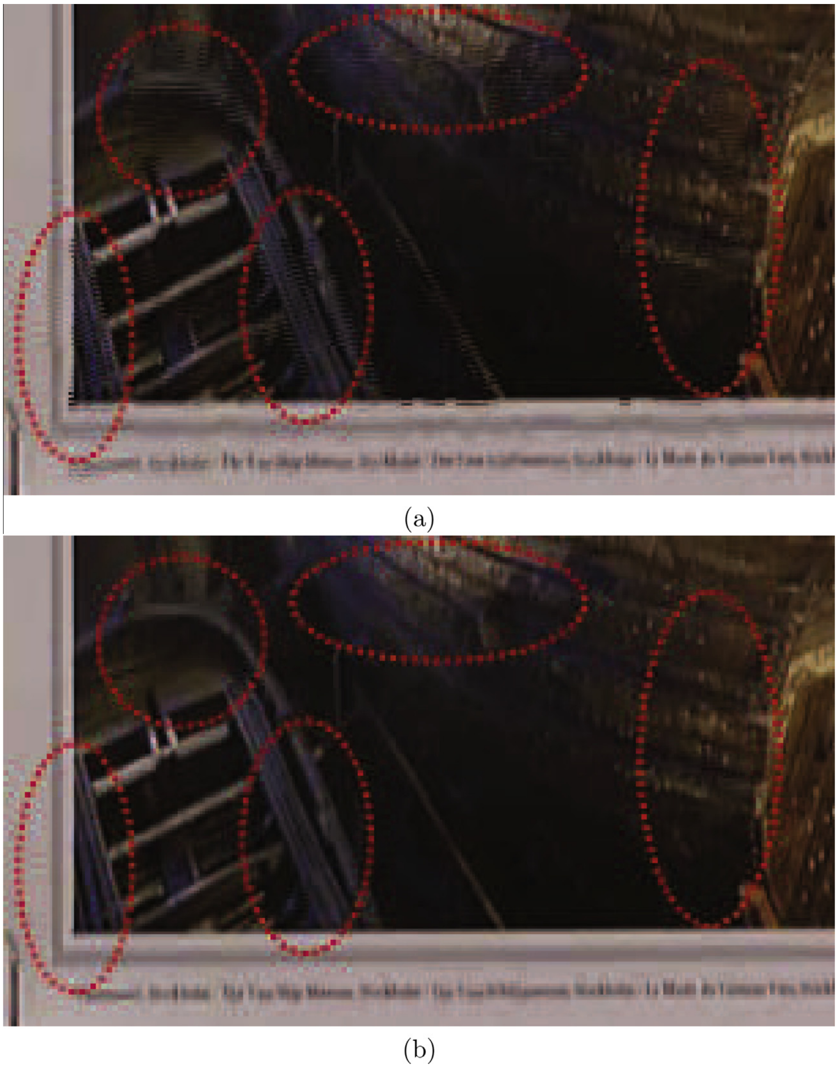
Table 2
The quality effect of the proposed strategy in terms of average PSNR (dB).

Strategy	QCIF videos		CIF videos		SD videos	full HD videos
	128 kb/s	256 kb/s	768 kb/s	1 Mb/s	4 Mb/s	20 Mb/s
Initiation	24.29	24.15	24.69	24.76	28.70	36.00
QPMRC	26.57	26.59	27.21	27.43	32.19	37.63
ASRF	30.25	31.68	29.75	30.27	33.93	37.90
MCLFQF	30.33	31.80	30.04	30.60	34.68	38.46
MBPR	N/A	N/A	N/A	N/A	N/A	38.54

residual block. In fact, when comparing with the computational time required in the de-interlacing process, the computational time required in decoding the inter-coded block dominates the computational bound. On the other hand, the computational overhead caused in the proposed de-interlacing process can be negligible when considering the computational effort in the decoding process for realizing the necessary SEs. This is the reason why the computational time performance of our proposed method is quite competitive to Dong and Ngan's method. As shown in Table 3, experimental results confirm it.

3. Experimental results

We compared the proposed de-interlacing method with the state-of-the-art de-interlacing method by Dong and Ngan [7]. We used both the real interlaced video sequences and the interlaced video sequences generated by removing the lines of progressive videos for comparisons. The three test real interlaced video sequences with the resolutions of 576i and 1080i are shown in Fig. 5. The eighteen test progressive video sequences with the resolutions of CIF, QCIF, SD, and full HD are listed in the second column of Table 1; in the experiments, the eighteen test progressive video sequences were first removed the corresponding lines to generate the interlaced video sequences. All the real and generated interlaced video sequences were encoded by H.264/AVC. The bitrates considered in the encoding process were 128 kb/s and 256 kb/s for the QCIF resolution, 768 kb/s and 1 Mb/s for the CIF resolution, 4 Mb/s for the SD and 576i resolutions, and 20 Mb/s for the full HD and 1080i resolutions. At decoding side, for each decoded MB, we performed, respectively, Dong and Ngan's de-interlacing method and the proposed de-interlacing method to fill the missing lines. All the experiments were implemented on the IBM compatible computer with Intel Core i7-960 CPU 3.2 GHz, 24 GB RAM, and Microsoft Windows 7 64 bit operating system. The compression platform was JM 17.2 [19], which was realized in Visual C++ 2008. All the experimental results are available in [18].



(a)

(b)

Fig. 8. For the de-interlaced Vasa ship calendar sequence, the magnified subframes generated by (a) Dong and Ngan's method and (b) the proposed method.

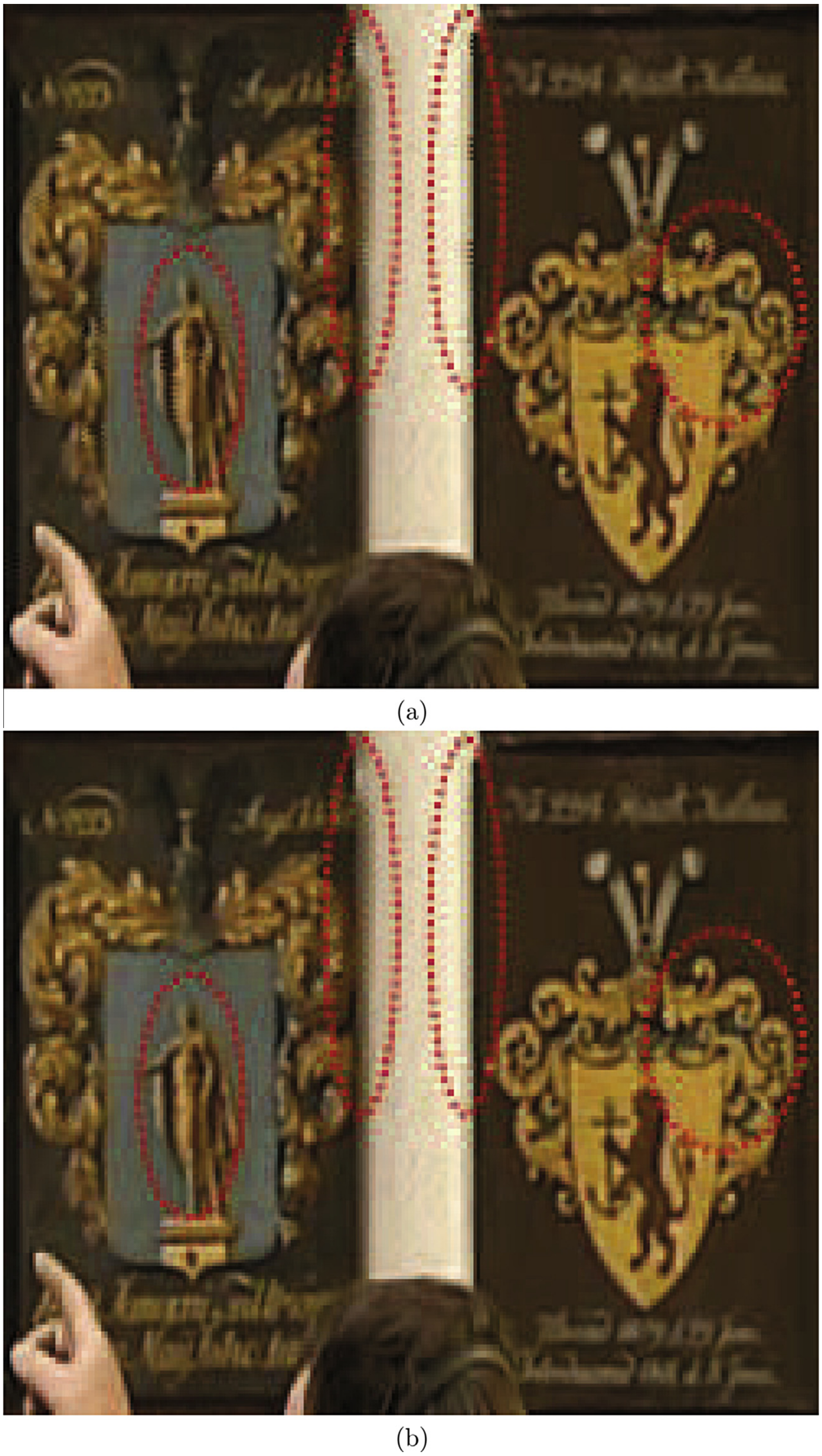
3.1. Quality comparison based on the generated interlaced video sequences

Based on the generated interlaced video sequences, the proposed de-interlacing method is first compared with Dong and Ngan's de-interlacing method in terms of the objective quality measure, PSNR. The PSNR of a de-interlaced video sequence with K frames, each with size $M \times N$, can be expressed as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{KMN}\sum_{k=1}^{K}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[F^k(i,j) - \widehat{F}^k(i,j)\right]^2}, \tag{11}$$

where $F^k(i,j)$ and $\widehat{F}^k(i,j)$ denote, respectively, the gray value of the pixel at position (i,j) in the k-th frame of the original and de-interlaced video sequences. Higher values of the PSNR indicate better quality of the de-interlaced video sequences. Table 1 gives the comparison results in terms of the PSNR values. It is clear that on average, the proposed de-interlacing method delivers better de-interlaced video quality than Dong and Ngan's method. The average quality improvement of the proposed method over Dong and Ngan's method for the video sequences with the resolutions of QCIF, CIF, SD, and full HD can achieve, respectively, more than 1.86, 1.33, 0.59, and 0.88 dBs.

Besides the quality superiority in terms of the objective metrics, we further demonstrate the subjective visual benefit of the proposed de-interlacing method. For inter-coded MBs, we present four new strategies to improve the quality of de-interlaced videos. The 391st frame with complex texture regions taken from City video sequence is first used to show the visual benefit of the proposed de-interlacing process for inter-coded MBs. Based on the area surrounded by the blue dotted line in Fig. 6(a) of the original 391st image frame of City video sequence with resolution of SD, Fig. 6(b)–(d) shows the magnified subframes of the original frame and the corresponding de-interlaced subframes generated by Dong and Ngan's method and the proposed method, respectively. It is obvious that the proposed method yields better visual quality than Dong and Ngan's method, especially in the striated regions surrounded by the red dotted lines. Then, we take the 12th frame Foreman video sequence to demonstrate the visual benefit of the proposed de-interlacing method. Based on the subframe surrounded by the blue dotted line in Fig. 7(a) of the original 12th image frame of Foreman video sequence, Fig. 7(b)–(d) shows the magnified subframes of the original frame and the corresponding de-interlaced subframes generated by Dong and Ngan's method and the proposed method, respectively. From Fig. 7, it is obvious that for the edge



(a)

(b)

Fig. 9. For the de-interlaced Shields video sequence, the magnified subframes generated by (a) Dong and Ngan's method and (b) the proposed method.

area surrounded by the red dotted line, the proposed method produces less jagged effect, implying better visual quality of the de-interlaced frame.

Next, we discuss the effect of the proposed new strategies on the de-interlaced video sequence quality in terms of PSNR. For de-interlacing inter-coded MBs, as mentioned before, we propose four new strategies, namely the quarter-pixel precision-based motion and residual compensations (QPMRC), the adaptive selection of reference frame (ASRF) to reduce error propagation, the motion compensation- and low-pass filter-based quality refinement (MCLFQF), and the motion blur propagation reduction (MBPR) for full HD video sequences. In the experiments, the inter-coded MBs were initially de-interlaced by the motion compensation with the integer-pixel precision MV, and then equipped each new strategy gradually. Table 2 gives the quality effect of each proposed strategy on average PSNRs of the de-interlaced video sequences. It is observed that the video quality in terms of PSNR increases gradually with the increase of equipping the proposed new strategies, implying the proposed four strategies have positive effect on the de-interlaced video quality.

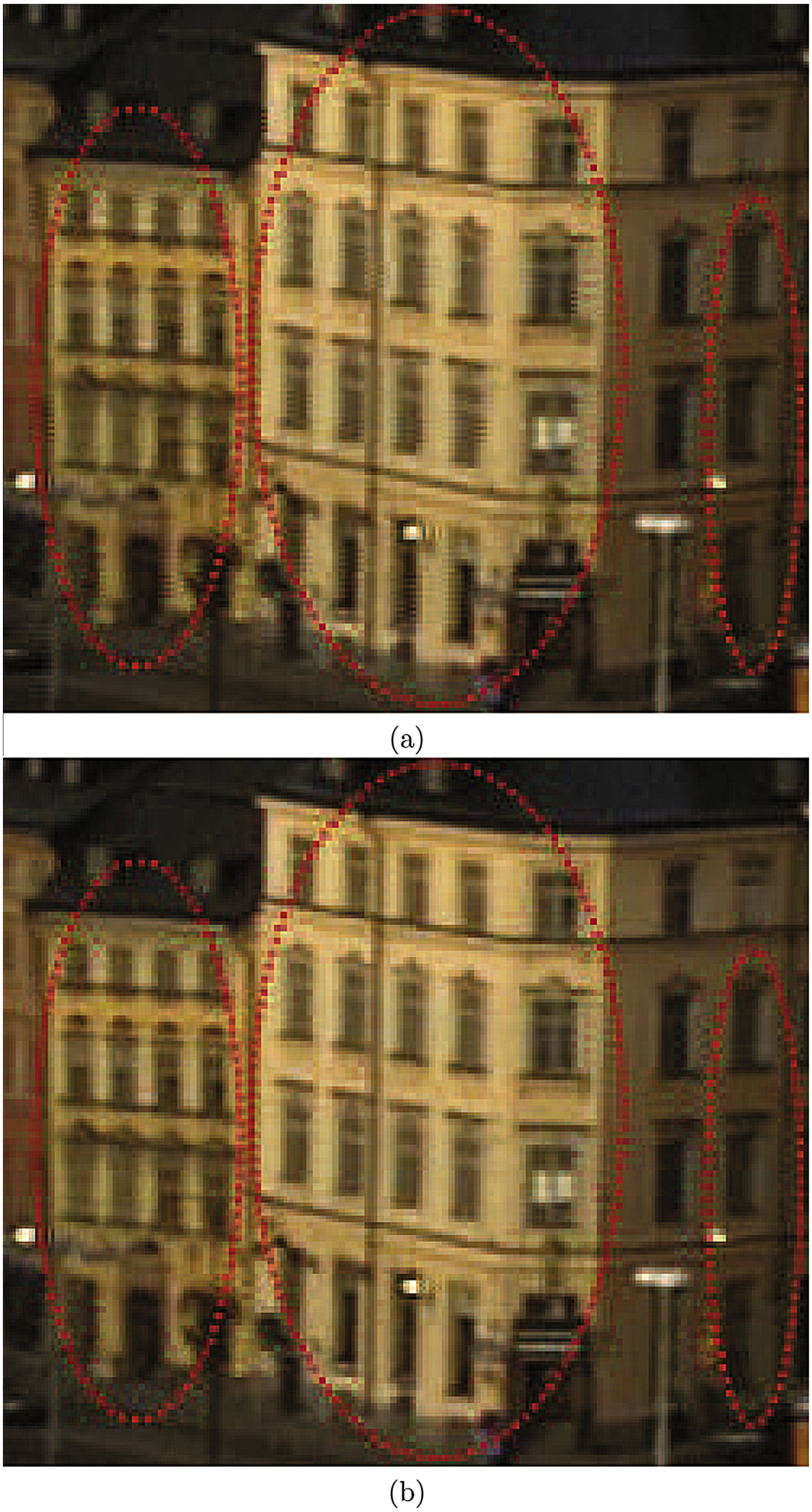


Fig. 10. For the de-interlaced Stockholm video sequence, the magnified subframes generated by (a) Dong and Ngan's method and (b) the proposed method.

3.2. Quality comparison based on the real interlaced video sequences

Apart from the comparisons based on the generated interlaced video sequences, the subjective visual evaluation based on the real interlaced video sequences shown in Fig. 5 is used to demonstrate the effectiveness of the proposed de-interlacing method. De-interlacing often generates the aliasing artifacts on texture regions and results in unpleasing visual perception. Magnified subframes with detailed textures taken from the image frames of the de-interlaced video sequences were used for visual comparison. For the 199th image frame of the de-interlaced Vasa ship calendar sequence, Fig. 8(a) and (b) shows the magnified subframes produced by Dong

Table 3
The average de-interlacing time (millisecond/frame) for Dong and Ngan's method and the proposed method.

Resolution	Sequence	Dong and Ngan's method		The proposed method	
QCIF		128 kb/s	256 kb/s	128 kb/s	256 kb/s
	Akiyo	8.22	8.37	8.23	8.49
	Carphone	8.35	8.58	8.32	8.56
	Coastguard	8.33	8.58	8.34	8.69
	Foreman	8.36	8.71	8.34	8.69
	Mobile	8.36	8.95	8.45	8.89
	Average	8.32	8.64	8.34	8.66
CIF		768 kb/s	1 Mb/s	768 kb/s	1 Mb/s
	Bus	32.28	32.64	31.89	33.23
	Coastguard	32.14	32.61	32.77	32.73
	Flower	32.22	32.65	32.67	32.62
	Football	31.93	32.18	31.98	32.31
	Foreman	32.25	33.98	32.82	33.47
	News	31.73	31.98	31.63	31.90
	Average	32.09	32.67	32.29	32.71
SD		4 Mb/s			
	City	132.91		134.55	
	Ice	129.66		132.59	
	Soccer	131.50		133.85	
	Average	131.36		133.66	
full HD		20 Mb/s			
	Pedestrian area	688.72		689.81	
	Rush hour	696.16		701.82	
	Tractor	685.93		700.22	
	River bed	665.51		693.51	
	Average	684.08		696.34	
576i		4 Mb/s			
	Vasa ship calendar	132.73		133.26	
	Shields	132.98		132.96	
	Average	132.86		133.11	
1080i		20 Mb/s			
	Stockholm	679.36		684.89	

and Ngan's method and the proposed method, respectively. It is clear that for the texture regions circled by the red dotted lines, the proposed method yields fewer aliasing artifacts than Dong and Ngan's method, indicating that the proposed de-interlacing method can produce better visual effect on the de-interlaced video sequence.

Similarly, we utilize the magnified subframes taken from the 113th image frames of the de-interlaced Shields video sequences produced, respectively, by Dong and Ngan's method and the proposed method for evaluation, as shown in Fig. 9. It is clear that the proposed de-interlacing method delivers better anti-aliasing effect and hence better visual effect than Dong and Ngan's method, especially in the texture regions circled by the red dotted lines. Further, for the high resolution Stockholm video sequence, similar comparison result based on the 105th image frames of the de-interlaced video sequences is shown in Fig. 10. In [18], readers can refer to more visual benefits of the proposed method over Dong and Ngan's method.

3.3. Computational time comparison

To demonstrate the feasibility of a de-interlacing method, the computational time evaluation should be addressed. This sub-section gives the computational time comparison between Dong and Ngan's method and the proposed method. Specifically, we evaluated the average computational time spent in reconstructing one single frame of a de-interlaced video sequence. Table 3 gives the related average computational time performance comparison. In accordance with the computational analysis in Section 2.6, the computational time performance of the proposed de-interlacing method is competitive to Dong and Ngan's method. Even though for the full HD video sequences shown in Table 3, the gap of average computational time between two de-interlacing methods is only 12.26 ms/frame, implying that the proposed de-interlacing method is effective and feasible.

4. Conclusion

We have presented an efficient de-interlacing method for H.264-coded video sequences with different resolutions. Based on the SEs in H.264 bitstreams, for inter-coded MBs, four new strategies, namely (1) the quarter-pixel precision-based motion and residual compensations, (2) the adaptive selection of reference frame to reduce error propagation, (3) the motion compensation- and low-pass filter-based quality refinement, and (4) the motion blur propagation reduction for full HD video sequences, improve the quality of de-interlaced videos as well as alleviates the error propagation side effect, forming the main contribution of this paper. Experimental results on the real and generated interlaced video sequences with various resolutions demonstrate that the proposed de-interlacing method delivers better de-interlaced video quality in terms of both the objective video quality measure and subjective visual effect when compared with the method by Dong and Ngan. Based on the generated interlaced video sequences, the average PSNR quality improvement of the proposed method over Dong and Ngan's method for the de-interlaced video sequences with the resolutions of QCIF, CIF, SD, and full HD can achieve, respectively, more than 1.86, 1.33, 0.59, and 0.88 dBs.

Based on the real interlaced video sequences, the proposed method delivers better anti-aliasing effect and hence better visual effect than Dong and Ngan's method. Although the proposed strategy on selecting the reference frame adaptively can reduce the error propagation when the current field and the reference field are both top fields or bottom fields, it is still a research issue for the case that the current field and the reference field are different types of fields. Another research issue is to apply the proposed method to those interlaced sequences in YUV format with rough motions. It is worth noting that the computational time performance of our proposed method is quite competitive to Dong and Ngan's method and the experimental results confirm it.

References

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC,) Joint Video Team of ISO/IEC and ITU-T, 2003.

[2] Y.L. Chang, S.F. Lin, C.Y. Chen, L.G. Chen, Video de-interlacing by adaptive 4-field global/local motion compensated approach, IEEE Trans. Circuits Syst. Video Technol. 15 (12) (2005) 1569–1582.

[3] J. Chang, Y.D. Kim, G.S. Shin, M.G. Kang, Adaptive arbitration of intra-field and motion compensation methods for de-interlacing, IEEE Trans. Circuits Syst. Video Technol. 19 (8) (2009) 1214–1220.

[4] Y.R. Chen, S.C. Tai, True motion-compensated de-interlacing algorithm, IEEE Trans. Circuits Syst. Video Technol. 19 (10) (2009) 1489–1498.

[5] B.D. Choi, J.W. Han, C.S. Kim, S.J. Ko, Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation, IEEE Trans. Circuits Syst. Video Technol. 17 (4) (2007) 407–416.

[6] P. Delogne, L. Cuvelier, B. Maison, B. van Caillie, L. Vandendorpe, Improved interpolation, motion estimation and compensation for interlaced pictures, IEEE Trans. Image Process. 3 (5) (1994) 482–491.

[7] J. Dong, K.N. Ngan, Real-time de-interlacing for H.264-coded HD videos, IEEE Trans. Circuits Syst. Video Technol. 20 (8) (2010) 1144–1149.

[8] T. Doyle, M. Looymans, Progressive scan conversion using edge information, in: L. Chairiglione (Ed.), Signal Processing of HDTV II, Elsevier, Amsterdam, The Netherlands, 1990, pp. 711–721.

[9] Y.C. Fan, C.H. Chung, De-interlacing algorithm using spatial–temporal correlation-assisted motion estimation, IEEE Trans. Circuits Syst. Video Technol. 19 (7) (2009) 932–944.

[10] G. de Haan, E.B. Bellers, De-interlacing: an overview, Proc. IEEE 86 (9) (1998) 1839–1857.

[11] R. Li, B. Zeng, M.L. Liou, Reliable motion detection/compensation for interlaced sequences and its applications to deinterlacing, IEEE Trans. Circuits Syst. Video Technol. 10 (1) (2000) 23–29.

[12] M. Park, M. Kang, K. Nam, S. Oh, New edge dependent deinterlacing algorithm based on horizontal edge pattern, IEEE Trans. Consum. Electronics 49 (4) (2003).

[13] M. Renfors, T. Huunhtanen, A. Nieminen, T. Koivunen, Linear and non-linear filters for sampling structure conversion of two-dimensional sequences, in: L. Chiariglione (Ed.), Signal Processing of HDTV, Elsevier, Amsterdam, 1990, pp. 685–694.

[14] R.J. Schutten, G. de Haan, Real-time 2-3 pull-down elimination applying motion estimation/compensation on a programmable device, IEEE Trans. Consum. Electron. 44 (3) (1998) 930–938.

[15] S.C. Tai, C.S. Yu, F.J. Chang, A motion and edge adaptive deinterlacing algorithm, in: Proc. of 2004 IEEE International Conference on Multimedia and Expo (ICME'04), Taipei, Taiwan, vol. 1, 2004, pp. 659–662.

[16] C.H. Wu, S.C. Tai, De-interlacing Algorithm Using Motion Adaptive and Edge-based Analysis, VDM Verlag, Saarbrucken, Germany, 2008.

[17] H. Yoo, J. Jeong, Direction-oriented interpolation and its application to de-interlacing, IEEE Trans. Consum. Electron. 48 (4) (2003) 954–962.

[18] Available: <http://140.118.175.164/WJYang/paper/H264DeInterlacing/>.

[19] Available: <http://iphome.hhi.de/suehring/tml/download/old_jm/>.