Signal Processing: *Image Communication* 28 (2013) 254–266

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

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

Up-sampling oriented frame rate reduction

Yongbing Zhang [a,*], Haoqian Wang [a], Debin Zhao [b]

[a] *Shenzhen Key Laboratory of Broadband Network & Multimedia, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China*
[b] *Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China*

ARTICLE INFO

Article history:
Received 5 March 2012
Accepted 12 December 2012
Available online 23 December 2012

Keywords:
Frame rate reduction
Temporal up-sampling
Motion compensation frame interpolation

ABSTRACT

Reducing the frame rate of a sequence results in the information loss of interpolated frames that cannot be recovered by temporal up-sampling. The most prevalent frame rate reduction method is the direct down-sampling, which maintains the original frames of the input sequence with a constant interval. It achieves wide applications due to simplicity. However, it neglects the influences on interpolated frames brought by temporal up-sampling. To maintain more information during down-sampling, which will be used for interpolation, this paper proposes an up-sampling oriented frame rate reduction by hinging the temporal up-sampling and down-sampling. It is designed with the goal of improving the quality of up-sampled frames while keeping the down-sampled sequence faithful to the input one. For a particular temporal up-sampling method, e.g., motion compensation frame interpolation, the optimal down-sampled frame is the one that not only minimizes the difference between the original and the down-sampled frames but also minimizes the difference between the original and the corresponding up-sampled frames. Users can make a tradeoff between the two difference terms by selecting a proper lagrange factor depending on the applications. Furthermore, to decrease the space and computational complexity, a block-wise implementation is also devised in this paper. Experimental results conducted on various sequences demonstrate that the proposed frame rate reduction is able to significantly improve the quality of up-sampled frames compared with other down-sampling methods.

1. Introduction

In many video applications, the spatiotemporal resolution of a video sequence is often reduced to meet the severe bit rate requirement when transmitting over band width limited channels. For instance, the frame rate of the input sequence can be reduced to the half or even smaller by skipping or deleting frames before compression, and then the temporal resolution is restored via up-sampling (usually termed as frame rate up conversion in the temporal domain) at the decoder side. The quality of up-sampled frames highly depends on the performance of frame rate up conversion.

Numerous algorithms have been developed to address the temporal resolution improvement of a video sequence. The most prevalent one is the motion compensation frame interpolation (MCFI) [1], which interpolates one intermediate frame by motion compensation. MCFI places a high demand on the accuracy of motion vector of each block. Many pioneering works on the motion vector derivation for each block to be interpolated were reported in [1–6]. In [2,3], block matching algorithm (BMA) was utilized to derive the motion vector of each block in the to-be-interpolated frame. Since the motion vectors derived by BMA are often not accurate enough, several approaches for more faithful motion vector derivation have also been proposed in recent works [4–6]. Haan et al. [6]

* Corresponding author.
E-mail addresses: ybzhang@tsinghua.edu.cn (Y. Zhang), wanghaoqian@tsinghua.edu.cn (H. Wang).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2012.12.004

proposed a 3D recursive search (3DRS) method to obtain accurate motion vectors. Choi et al. [1] proposed a MCFI algorithm using bi-directional motion estimation to derive more faithful motion vectors. A hierarchical motion compensation technique was also proposed in [4] to achieve better visual quality. Both these two methods in [1,4] outperform BMA. However, they are both based on the assumption of translational motion with constant velocity, which is not always true. In [5], constant acceleration was exploited to derive more reliable motion trajectories. However, the assumption of constant acceleration does not always hold for all the regions, e.g., for the regions of non-rigid objects.

One significant difference between frame rate up conversion and video coding [7] is that the motion vector derivation process in the former is performed with the absence of actual frames. Hence, the derived motion trajectory may be not consistent sometimes. To resolve such a problem, many motion vector post-processing methods are proposed to smooth the motion field and improve the subjective perception [2,5,8]. What is more, for the areas with small objects, irregular shaped objects and object boundaries, fixed size block motion compensation usually does not work well. To deal with this situation, variable size block motion compensation was proposed to reconstruct the edge information with higher quality [9]. Furthermore, overlapped block motion compensation (OBMC) [10,11] is applied to suppress the blocking artifacts, which are usually observed when a block has a significantly different motion vector compared with its neighboring blocks. However, OBMC may sometimes over-smooth the edges of the image and thus degrades the image quality. To reduce the over-smooth effect of OBMC, Choi et al. [9] proposed an adaptive OBMC (AOBMC), in which OBMC coefficients were adjusted according to the reliability of neighboring motion vectors. However, AOBMC still has poor ability to represent some complex motions, such as zooming, rotation, and local deformations. To better capture the varying properties of local regions, auto regression model was also utilized to perform frame rate up conversion in [12,13].

Actually, the quality of up-sampled frames depends on not only the performance of frame rate up conversion method but also the information maintained in the down-sampled video sequence. In the traditional direct down-sampling method, the down-sampled sequence consists of frames with a constant interval within the input sequence. On the contrary, the down-sampled frame is adaptively selected from several frames so as to minimize inter-frame prediction error in [14,15]. However, these methods only suit for high frame rate video, since down-sampling with non-constant intervals for low frame rate video would yield jerky motion, i.e., perceivable discontinuity in the optical field. Besides, motion-compensated temporal filtering (MCTF) [16] can also be utilized to perform temporal down-sampling.

All these methods are able to keep the down-sampled sequences faithful to the input one to a great extent, but none of them consider the influences on interpolated frames brought by temporal up-sampling during down-sampling. To address this problem, an up-sampling oriented frame rate reduction, which hinges the temporal up-sampling and down-sampling, is proposed in this paper. The goal of the proposed algorithm is to improve the quality of up-sampled frames while keeping the down-sampled frames faithful to the original ones. In other words, for a particular temporal up-sampling method, the optimal down-sampled frame is the one that not only minimizes the difference between the original and down-sampled frames but also minimizes the difference between the original and the corresponding up-sampled frames. Users can make a tradeoff between the fidelity of the two error terms by selecting a proper Lagrange factor depending on the applications. To decrease the space and computational complexity of the proposed algorithm, a block-wise implementation of the proposed down-sampling method is also provided. Experimental results conducted on various sequences demonstrate that the proposed frame rate reduction algorithm is able to significantly improve the quality of the up-sampled frames compared with other down-sampling methods. Besides, when the proposed method is used for low bit-rate video compression, it can reduce the fluctuation between the decompressed frame and those interpolated by frame rate up conversion at the decoder side.

The rest of this paper is organized as follows. The problem formulation is first introduced in Section 2, where traditional MCFI is first reviewed and then the optimal solution of the up-sampling oriented frame rate reduction problem is derived. In Section 3, a block-wise implementation of the proposed down-sampling algorithm is given. The experimental results and analysis are provided in Section 4. Finally, this paper is concluded in Section 5.

2. Problem formulation

Fig. 1 illustrates the proposed up-sampling oriented frame rate reduction. It can be seen that the frame rate reduction is hinged to the temporal up-sampling because the up-sampling is considered when frame rate reduction is performed. In this section we will first give a brief review of traditional temporal up-sampling method, i.e., MCFI, and then the derivation of the optimal down-sampled frame will be presented.

2.1. Review of MCFI

MCFI usually interpolates one or several intermediate frames given two neighboring frames. Denote $X_t(\vec{n})$ as a pixel located at frame X_t with the discrete space coordinate $\vec{n}=(n_x,n_y)$ and its corresponding interpolated one is

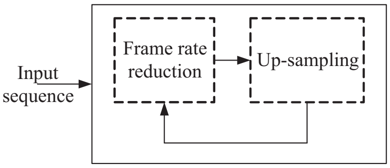

Fig. 1. The proposed up-sampling oriented frame rate reduction.

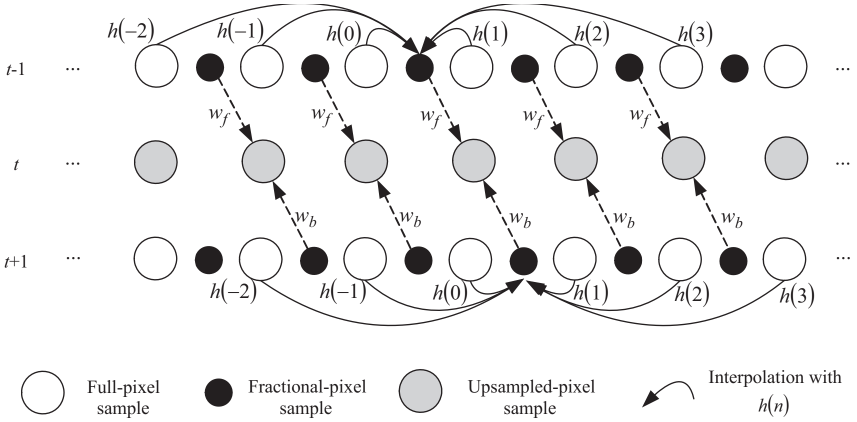

Fig. 2. 1:2 frame rate up conversion when $\vec{v}_f=(-1/2,0)$ and $\vec{v}_b=(1/2,0)$.

$\hat{X}_t(\vec{n})$. To capture the varying property of frame contents, in MCFI the whole frame is usually divided into a number of blocks S, and each block has a motion vector $\vec{v}=(v_x,v_y)$ with the horizontal component $v_x$and vertical component v_y, respectively. And then $X_t(\vec{n})$ can be formulated as

$$\begin{aligned}\hat{X}_t(\vec{n}) &= w_f P_f(\vec{n})+w_b P_b(\vec{n})\\ &= w_f X_{t-1}(\vec{n}+\vec{v}_f)+w_b X_{t+1}(\vec{n}+\vec{v}_b)\end{aligned} \tag{1}$$

where w_f and w_b are the relative weights of the forward predicted block P_f and the backward predicted block P_b, $\vec{v}_f$ and $\vec{v}_b$ represent the motion vectors in the forward and backward reference frames. For the majority cases, $w_f+w_b=1$ and $w_f=w_b=1/2$. More generally, $\vec{v}_f$ and $\vec{v}_b$ may be any fractional numbers [17]. If the motion vectors are of sub-pixel accuracy, Eq. (1) is applied to the corresponding references with fractional-pixel accuracy to yield the up-sampled signals accordingly.

When a finite impulse response (FIR) filter with 2M-tap is used for the 2-D separate interpolation, the reference signals with motion vectors of horizontally, vertically and diagonally half-pixel accuracy in each prediction direction can be yielded by

$$P(\vec{n})=\sum_{u=-M+1}^{M} h(u)X_r(n_x+\lfloor v_x\rfloor+u, n_y+\lfloor v_y\rfloor) \tag{2}$$

$$P(\vec{n})=\sum_{u=-M+1}^{M} h(u)X_r(n_x+\lfloor v_x\rfloor, n_y+\lfloor v_y\rfloor+u) \tag{3}$$

and

$$P(\vec{n})=\sum_{u_1=-M+1}^{M} h(u_1)\times\left(\sum_{u_0=-M+1}^{M} h(u_0)X_r(n_x+\lfloor v_x\rfloor+u_0, n_y+\lfloor v_y\rfloor+u_1)\right) \tag{4}$$

where $\lfloor\rfloor$ represents the operation rounded to the nearest integer pixel position towards minus infinity and $h(u)$ represents the tap coefficient. The interpolated values at the horizontal and vertical half-pixel positions are obtained by applying a one-dimensional 2M-tap FIR filter horizontally and vertically using Eqs. (2) and (3), respectively. For the diagonally half-pixel position, one-dimensional 2M-tap FIR filter needs to be performed horizontally firstly and then vertically using Eq. (4). The half pixels and full pixels are then utilized to interpolate the quarter-pixels via bilinear method. Fig. 2 illustrates the 1:2 frame rate up conversion process with horizontally half-pixel accuracy in both directions when a FIR filter h with 6-tap is used. The corresponding interpolations are first used to generate the forward and backward prediction blocks P_f and P_b using Eq. (2), respectively. And then the up-sampled pixel can be yielded by Eq. (1).

2.2. Derivation of the optimal down-sampled frame

Traditional MCFI usually tries to find the most faithful motion vectors for each block to be interpolated. Actually, it is easy to observe from Eq. (1) that the quality of up-sampled frames depends on not only the accuracy of motion vectors but also the information contained in the forward and backward reference frames. More information about the frame to be interpolated embedded in the forward and backward reference frames, up-sampled frames with much higher quality can be obtained. To transfer more information about the frame to be interpolated to the down-sampled frames, an up-sampling oriented frame rate reduction is proposed in this subsection. Here, we will take MCFI [1] as an example to describe the derivation of the optimal down-sampled frame, and it can be easily extended to other frame rate up conversion algorithms.

Define $\mathbf{X}_t$ as the original frame in the input video at time instance t in a vector form and the corresponding up-sampled frame is $\hat{\mathbf{X}}_t$. For simplicity, we will take 1:2 MCFI as an example to derive the optimal solution of the frame rate reduction problem. And of course, it can be easily extended to arbitrary ratio MCFI. The goal of the proposed frame rate reduction is to generate a high quality interpolated frame while at the same time make the down-sampled sequence faithful to the input one. Consequently, the optimal down-sampled frame should satisfy

$$\mathbf{X}^*_{t+1}=\min_{\mathbf{X}^c_{t+1}}\ ||\mathbf{X}_t-\hat{\mathbf{X}}_t||^2 \quad s.t.||\mathbf{X}_{t+1}-\mathbf{X}^c_{t+1}||^2<\tau \tag{5}$$

where $\mathbf{X}^*_{t+1}$ represents the optimal down-sampled frame, and $\mathbf{X}^c_{t+1}$ represents an arbitrary candidate value of the down-sampled frame. According to Eq. (5), the objective function of the optimal down-sampled frame can be expressed as

$$\mathbf{X}^*_{t+1}=\min_{\mathbf{X}^c_{t+1}}\left\{||\mathbf{X}_t-\hat{\mathbf{X}}_t||^2+\lambda||\mathbf{X}_{t+1}-\mathbf{X}^c_{t+1}||^2\right\} \tag{6}$$

where λ is a constant to make a tradeoff between the two error terms.

Assume we have obtained the optimal down-sampled frame $\mathbf{X}^*_{t-1}$, the up-sampled frame at time instance t can be generated by reformulating Eq. (1) as

$$\hat{\mathbf{X}}_t=w_f\vec{\mathbf{P}}(\mathbf{X}^*_{t-1},\vec{\mathbf{V}}_f)+w_b\vec{\mathbf{P}}(\mathbf{X}^c_{t+1},\vec{\mathbf{V}}_b) \tag{7}$$

where $\vec{\mathbf{V}}_f$ and $\vec{\mathbf{V}}_b$ represent the motion field of the up-sampled frame in the forward and backward reference frames, $\vec{\mathbf{P}}(\mathbf{X}^*_{t-1},\vec{\mathbf{V}}_f)$ and $\vec{\mathbf{P}}(\mathbf{X}^c_{t+1},\vec{\mathbf{V}}_b)$ represent interpolating $\mathbf{X}^*_{t-1}$ and $\mathbf{X}^c_{t+1}$ to the corresponding predictions utilizing motion fields $\vec{\mathbf{V}}_f$ and $\vec{\mathbf{V}}_b$, respectively. Here the prediction generation function $\vec{\mathbf{P}}(\mathbf{X}^c_{t+1},\vec{\mathbf{V}}_b)$ can be expressed as

$$\vec{\mathbf{P}}(\mathbf{X}^c_{t+1},\vec{\mathbf{V}}_b)=\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1} \tag{8}$$

where $\mathbf{H}_{\vec{\mathbf{V}}_b}$ represents the interpolation matrix determined by the motion field $\vec{\mathbf{V}}_b$. It should be noted that $\mathbf{H}_{\vec{\mathbf{V}}_b}$, composed of coefficients of the FIR filter, can be specified as

$$\mathbf{H}_{\vec{\mathbf{V}}_b}=\begin{bmatrix} h_{0,0} & h_{0,1} & .. & h_{0,M\times N-1}\\ h_{1,0} & h_{1,1} & .. & h_{1,M\times N-1}\\ h_{2,0} & h_{2,1} & .. & h_{2,M\times N-1}\\ .. & .. & .. & ..\\ h_{M\times N-1,0} & h_{M\times N-1,1} & .. & h_{M\times N-1,M\times N-1}\end{bmatrix} \tag{9}$$

where M and N represent the height and width of each frame, $h_{k,l}$ represents the interpolation coefficient contributed by the lth full-pixel during the interpolation of the kth prediction pixel. Here, if $\vec{\mathbf{V}}_b$ locates at full-pixel position, $\mathbf{H}_{\vec{\mathbf{V}}_b}$ is composed of a series of 0 and 1, and if $\vec{\mathbf{V}}_b$ locates at fractional-pixel position, $\mathbf{H}_{\vec{\mathbf{V}}_b}$ is composed of corresponding FIR interpolation coefficients as shown in Eqs. (2–4).

Incorporate Eqs. (7) and (8) into Eq. (6), we have

$$\mathbf{X}^*_{t+1}=\min_{\mathbf{X}^c_{t+1}}J=\min_{\mathbf{X}^c_{t+1}}\left\{\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}^*_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1}-\mathbf{X}_t\|^2+\lambda\|\mathbf{X}_{t+1}-\mathbf{X}^c_{t+1}\|^2\right\} \tag{10}$$

It should be noted that when λ approaches the infinite, Eq. (10) equals to the direct down-sampling method, and when λ equals to 0, Eq. (10) only takes the influence of up-sampled frame into account.

Assuming the optimal down-sampled previous frame $\mathbf{X}^*_{t-1}$ equals to $\mathbf{X}_{t-1}$, Eq. (10) becomes

$$\mathbf{X}^*_{t+1}=\min_{\mathbf{X}^c_{t+1}}\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1}-\mathbf{X}_t\|^2 \tag{11}$$

when λ equals to 0. And the MCFI result of direct down-sampling can be expressed as

$$\begin{aligned}\hat{\mathbf{X}}_t&=w_f\vec{\mathbf{P}}(\mathbf{X}_{t-1},\vec{\mathbf{V}}_f)+w_b\vec{\mathbf{P}}(\mathbf{X}_{t+1},\vec{\mathbf{V}}_b)\\ &=w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}_{t+1}\end{aligned} \tag{12}$$

Since the optimal down-sampled frame $\mathbf{X}^*_{t+1}$ in Eq. (11) is the one has the minimal of $\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1}-\mathbf{X}_t\|^2$, we have

$$\begin{aligned}&\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^*_{t+1}-\mathbf{X}_t\|^2\\ &\quad\le\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}_{t+1}-\mathbf{X}_t\|^2\end{aligned} \tag{13}$$

which means that for the interpolated frame $\hat{\mathbf{X}}_t$, the proposed down-sampling method is able to achieve better performance than the direct down-sampling method. However, since MCFI utilizes the previous and following frames, the difference between down-sampled frame and the corresponding original one must be considered. Consequently λ cannot be set to zero to ensure the following up-sampled frames also have good performance, which will be verified by the first part in the following experiment section.

It is easy to obtain the derivative of Eq. (10) as

$$\begin{aligned}\frac{\partial J}{\partial\mathbf{X}^c_{t+1}}&=w_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\left[w_f\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}^*_{t-1}+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1}-\mathbf{X}_t\right]\\ &\quad+2\lambda[\mathbf{X}^c_{t+1}-\mathbf{X}_{t+1}]\\ &=w_fw_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}^*_{t-1}+w_b{}^2\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_b}\mathbf{X}^c_{t+1}\\ &\quad-w_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{X}_t+2\lambda\mathbf{X}^c_{t+1}-2\lambda\mathbf{X}_{t+1}\\ &=\left[w_b{}^2\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_b}+2\lambda\right]\mathbf{X}^c_{t+1}-w_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{X}_t-2\lambda\mathbf{X}_{t+1}\\ &\quad+w_fw_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}^*_{t-1}\end{aligned} \tag{14}$$

Set $(\partial J/\partial\mathbf{X}^c_{t+1})=0$, we have

$$\begin{aligned}\mathbf{X}^*_{t+1}&=\left[w_b{}^2\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_b}+2\lambda\right]^{-1}\\ &\quad\times\left[w_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{X}_t+2\lambda\mathbf{X}_{t+1}-w_fw_b\mathbf{H}_{\vec{\mathbf{V}}_b}{}^T\mathbf{H}_{\vec{\mathbf{V}}_f}\mathbf{X}^*_{t-1}\right]\end{aligned} \tag{15}$$

Since the proposed down-sampling is performed frame by frame, down-sampled frame $\mathbf{X}^*_{t-1}$ is already available when deriving $\mathbf{X}^*_{t+1}$. In this paper, direct down-sampled sequences are utilized to derive the motion filed of the up-sampled frame. There would be some mismatch when performing MCFI after the obtaining of the optimal down-sampled frames. One solution is to perform the down-sampling iteratively, where the optimal down-sampled frames in the previous iteration are utilized to obtain the motion field and then perform down-sampling in the current iteration utilizing the motion filed obtained in the current iteration. However, we find that only one iteration time, as the method in this paper, will be good enough. In Eq. (15), matrix $w_b^2\mathbf{H}^T_{\vec{\mathbf{V}}_b}\mathbf{H}_{\vec{\mathbf{V}}_b}+2\lambda$ can be reversed for the majority cases. However, when $w_b^2\mathbf{H}^T_{\vec{\mathbf{V}}_b}\mathbf{H}_{\vec{\mathbf{V}}_b}+2\lambda$ has no inverse, $\mathbf{X}_{t+1}$ will be considered as the final down-sampled frame $\mathbf{X}^*_{t+1}$.

It should be noted that the motion vector field $\vec{\mathbf{V}}_b$ may be varying within one fame and the dimension of $\mathbf{H}_{\vec{\mathbf{V}}_b}$ is too high. For example, the dimension of $\mathbf{H}_{\vec{\mathbf{V}}_b}$ is $MN\times MN$, as indicated in Eq. (9), which is too demanding in terms of storage and computation complexities. To reduce the computational and space complexity and better capture the motion characteristics of different regions, a block-wise implementation is proposed in the following section.

3. Block-wise implementation

As mentioned in the previous section, each block in MCFI has its own motion vector, by which motion compensation is performed. In the proposed frame rate reduction, the motion vector of each block can also be utilized to construct its corresponding interpolation matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}$ as shown in Eq. (9). Define $\mathbf{X}_t(q)$ as the qth block, whose size is $m\times n$, within frame $\mathbf{X}_t$, and we will take $\mathbf{X}_t(q)$ as an example to describe the proposed block-wise implementation. If $\vec{\mathbf{V}}_b(q)$ is of full-pixel accuracy, it will not necessarily perform the sub-pixel interpolation in Eq. (2) to Eq. (4). Consequently, the elements along the diagonal direction of interpolation matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}$ will be 1, and all the remaining elements are 0. In other words, $\mathbf{H}_{\vec{\mathbf{V}}_b}$ becomes the identity matrix, and the optimal solution in Eq. (15) will be

$$\mathbf{X}^*_{t+1}(q)=\frac{w_b\mathbf{X}_t(q)+2\lambda\mathbf{X}_{t+1}(q)-w_f w_b\mathbf{X}^*_{t-1}(q)}{{w_b}^2+2\lambda}$$

However, it will be more complicated if $\vec{\mathbf{V}}_b(q)$ is of fractional-pixel accuracy, since the fractional-pixel is interpolated from the nearby full pixels using FIR or bilinear interpolation method. Enough attention should be paid to the calculation of interpolation matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$, especially for the elements corresponding to the boundary pixels, since it will impose great influence on the performance. Obviously, the interpolation of the majority prediction pixels only involves full-pixels within the same block, so long as all the involved full-pixels are located inside the current block. However, for the boundary pixels, some full-pixels outside the current block may be also involved, e.g., the interpolation of the pixels indicated by the black circles in Fig. 3 involves the full-pixels outside the current block indicated by the gray circles in Fig. 3, due to the FIR interpolation. If we represent the optimal values of all the pixels within the current block as a variable of column vector, which will be derived by the proposed frame rate reduction algorithm, other unknown variables (the full-pixels, which are located outside the current block and are involved during the FIR interpolation of boundary pixels within the current block) will be involved inevitably.

To tackle such a problem, we define $\mathbf{C}_q$ as a constant vector, and the prediction generation function of Eq. (8) can be reformulated as

$$\vec{\mathbf{P}}(\mathbf{X}^*_{t+1}(q),\vec{\mathbf{V}}(q))=\mathbf{H}_{\vec{\mathbf{V}}_b}(q)\mathbf{X}^*_{t+1}(q)+\mathbf{C}_q \tag{16}$$

where $\mathbf{C}_q$ represents the value contributed by the pixels outside the current block during interpolation and $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ represents the interpolation matrix of the current block determined by the motion vector $\vec{\mathbf{V}}_b(q)$. It is noted that each element of $\mathbf{C}_q$ represents the accumulated summation of multiplications between the involved full-pixel outside the current block and its corresponding FIR coefficient. Here, the introduction of $\mathbf{C}_q$ is to efficiently utilize the information of full-pixels outside the current block, which is unavoidable due to the FIR interpolation. $\mathbf{C}_q$ can be first constructed before the derivation of current block begins.

Incorporating Eq. (16) into Eq. (10), the objective function to derive the optimal value of the qth block should be

$$\begin{aligned}J_q=&\left\|w_f\mathbf{H}_{\vec{\mathbf{V}}_f}(q)\mathbf{X}^*_{t-1}(q)+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}(q)\mathbf{X}^c_{t+1}(q)\right.\\&\left.+\mathbf{C}_q-\mathbf{X}_t(q)\right\|^2+\lambda\left\|\mathbf{X}_{t+1}(q)-\mathbf{X}^c_{t+1}(q)\right\|^2\end{aligned} \tag{17}$$

Based on Eq. (17), the partial derivative of J_q can be formulated as

$$\begin{aligned}\frac{\partial J_q}{\partial\mathbf{X}^c_{t+1}(q)}=&\,w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\\&\times[w_f\mathbf{H}_{\vec{\mathbf{V}}_f}(q)\mathbf{X}^*_{t-1}(q)+w_b\mathbf{H}_{\vec{\mathbf{V}}_b}(q)\mathbf{X}^c_{t+1}(q)+\mathbf{C}_q-\mathbf{X}_t(q)]\\&+2\lambda[\mathbf{X}^c_{t+1}(q)-\mathbf{X}_{t+1}(q)]=w_f w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{H}_{\vec{\mathbf{V}}_f}(q)\mathbf{X}^*_{t-1}(q)\\&+w_b^2\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{H}_{\vec{\mathbf{V}}_b}(q)\mathbf{X}^c_{t+1}(q)+w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{C}_q\\&-w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{X}_t(q)+2\lambda\mathbf{X}^c_{t+1}(q)-2\lambda\mathbf{X}_{t+1}(q)\end{aligned}$$

involved inevitably.

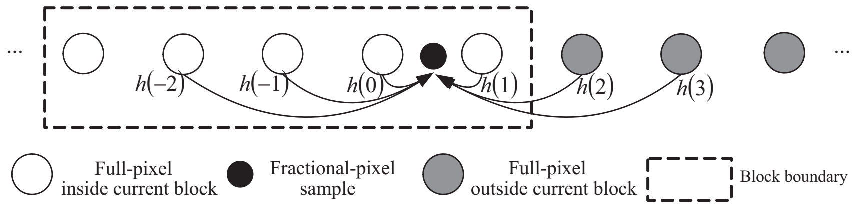

Fig. 3. Interpolations at block boundary.

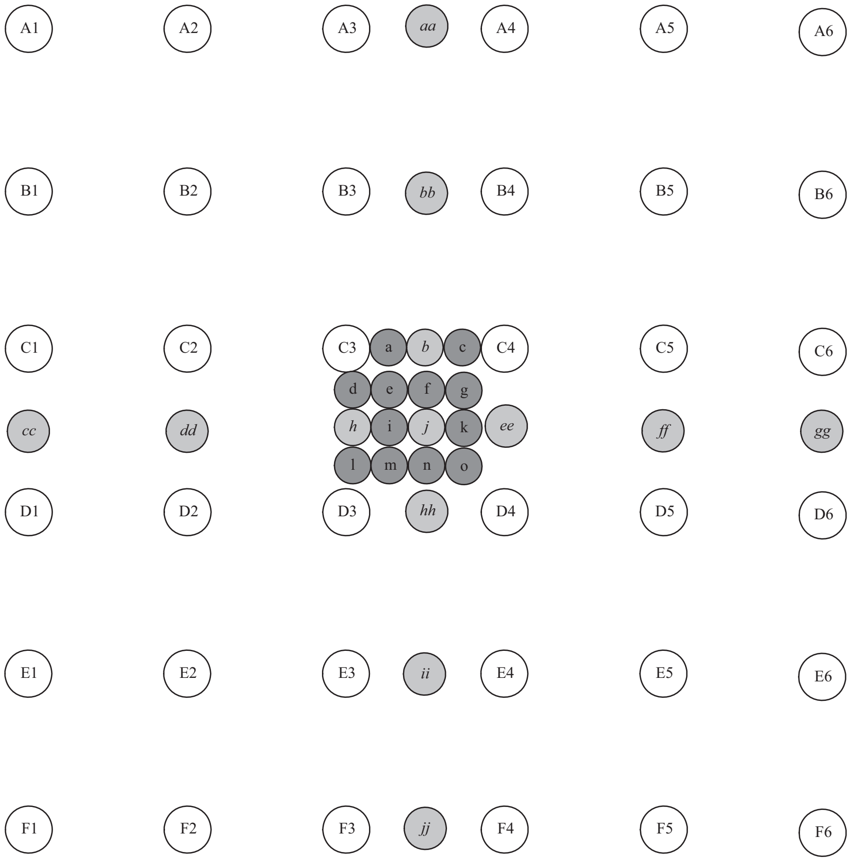

Fig. 4. Sample grid with quarter-pixel interpolation accuracy for signals (upper-case letters indicate samples on the full-pixel positions, lower-case italic letters indicate samples on the half-pixel positions and the remaining lower-case letters indicate samples on the quarter-pixel positions).

$$
\begin{aligned}
&= [w_b^2\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{H}_{\vec{\mathbf{V}}_b}(q)+2\lambda]\mathbf{X}^c_{t+1}(q)-w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{X}_t(q)\\
&-2\lambda\mathbf{X}_{t+1}(q)+w_f w_b\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{H}_{\vec{\mathbf{V}}_f}(q)\mathbf{X}^*_{t-1}(q)+w_b^2\mathbf{H}^T_{\vec{\mathbf{V}}_b}(q)\mathbf{C}_q
\end{aligned}
\tag{18}
$$

Set $(\partial J_q/\partial\mathbf{X}^c_{t+1}(q))=0$, we have

$$
\begin{aligned}
\mathbf{X}^*_{t+1}(q) &= \left[w_b{}^2\mathbf{H}_{\mathbf{V}_b}{}^T(q)\mathbf{H}_{\mathbf{V}_b}(q)+2\lambda\right]^{-1}\\
&\times\left[w_b\mathbf{H}_{\mathbf{V}_b}{}^T(q)\mathbf{X}_t(q)+2\lambda\mathbf{X}_{t+1}(q)\right.\\
&\left.-w_f w_b\mathbf{H}_{\mathbf{V}_b}{}^T(q)\mathbf{H}_{\mathbf{V}_f}(q)\mathbf{X}^*_{t-1}(q)-w_b{}^2\mathbf{H}_{\mathbf{V}_b}{}^2(q)\mathbf{C}_q\right]
\end{aligned}
\tag{19}
$$

It should be noted that the calculation of $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ and $\mathbf{C}_q$ will depend on the motion vector $\vec{\mathbf{V}}_b(q)$of the qth block. The fractional-pixel interpolation process applying 6-tap FIR filter is illustrated in Fig. 4, where the interpolation of fractional-pixel located at different positions will involve different full-pixels and use different coefficients. Consequently, we should carefully calculate varying values of $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ and $\mathbf{C}_q$ depending on the value of $\vec{\mathbf{V}}_b(q)$. Here, we will take the interpolation of half-pixel located at position j shown in Fig. 4 as an example to describe the calculation of $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ and $\mathbf{C}_q$.

Table 1
Calculation of interpolation matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}(q,j)$ for the half-pixel j.

```
Input: $H_j$, block size $m \times n$
Output: Matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}(q,j)$
for $k=0$ to $m \times n-1$
    for $l=0$ to $m \times n-1$
        $\mathbf{H}_{\vec{\mathbf{V}}_b}(q,j,k,l)=0$
    end for
end for
for $k=0$ to $m-1$
    for $l=0$ to $n-1$
        for $u=-2$ to 3
            $kk=k+u$
            for $v=-2$ to 3
                $ll=l+v$
                if $0 \le kk \le m-1 \& 0 \le ll \le n-1$
                    $\mathbf{H}_{\vec{\mathbf{V}}_b}(q,j,k \times n+l,kk \times n+ll)=H_j(u+2,v+2)$
                end if
            end for
        end for
    end for
end for
```

The interpolation of half-pixel j involves full-pixels A1, A2,…A6, B1, B2, …,B6, …F1, F2, …F6. We first derive the corresponding coefficient of each full-pixel as $H_j=h^T h$,

Table 2
Calculation of constant vector $\mathbf{C}_{qj}$ for the half-pixel j.

```
Input:$H_j$, block size $m \times n$, $\mathbf{X}_{t+1}$
Output:Vector $\mathbf{C}_{qj}$
for $k=0$ to $m \times n-1$
    $\mathbf{C}_{qj}(k)=0$
end for
for $k=0$ to $m-1$
    for $l=0$ to $n-1$
        sum$=0$
        for $u=-2$ to 3
            $kk=k+u$
            for $v=-2$ to 3
                $ll=l+v$
                if $kk<0|kk\geq m|ll<0||ll\geq n$
                    $sum+=H_j(u+2,v+2)\times \mathbf{X}_{t+1}(kk,ll)$ ;
                end if
            end for
        end for
        $\mathbf{C}_{qj}(k\times n+l)=sum$
    end for
end for
```

with $h=[h(-2),h(-1),..h(3)]$ denoting the FIR coefficients. The pseudo codecs of calculations of $\mathbf{H}_{\vec{\mathbf{v}}_b}(q)$ and $\mathbf{C}_q$ are given in Tables 1 and 2, respectively.

4. Experimental results and analysis

In this section, various experiments are conducted to demonstrate the validity of the proposed frame rate reduction algorithm. Each test sequence is first down-sampled, and then the down-sampled sequence is interpolated by MCFI [1], OBMC [11] and AOBMC [9]. In the proposed algorithm, we utilize the 6-tap FIR filter $(1, -5, 20, 20, -5, 1)/32$ as in H.264/AVC [7] and the block size is set to be 8×8.

4.1. Influence of trade-off parameter

In this subsection, the influence of trade-off parameter λ is discussed. The simulations under various values of λ over test sequences City (QCIF) and Coastguard (CIF) are carried out and evaluated. The first 150 frames of each test sequence are down-sampled by the proposed algorithm.

Fig. 5 depicts the curve between the average mean square error (MSE) and parameter λ over test sequences City (QCIF) and Coastguard (CIF). Here, not only the MSE between each down-sampled image $\mathbf{X}^*_{t+1}$ and the corresponding original frame $\mathbf{X}_{t+1}$, but also the MSE between each up-sampled frames $\hat{\mathbf{X}}_t$ and the corresponding original frame $\mathbf{X}_t$ is calculated. The MSE in Fig. 5 represents the average MSE of all the down-sampled and interpolated frames over the test sequences. It can be observed that MSE has the largest value when λ equals to 0. This is because the error term between the down-sampled frame and the corresponding original frame, as indicated in Eq. (10), is neglected. When λ equals to 1, MSE drops significantly, and MSE reaches its minimum value when λ equals to 2. This is greatly attributed to the joint consideration of both the down-sampled and interpolated frames. However, when λ exceeds 2, MSE begins to increase with the increase of λ. Based on this observation, λ is set to 2 in the following experiments.

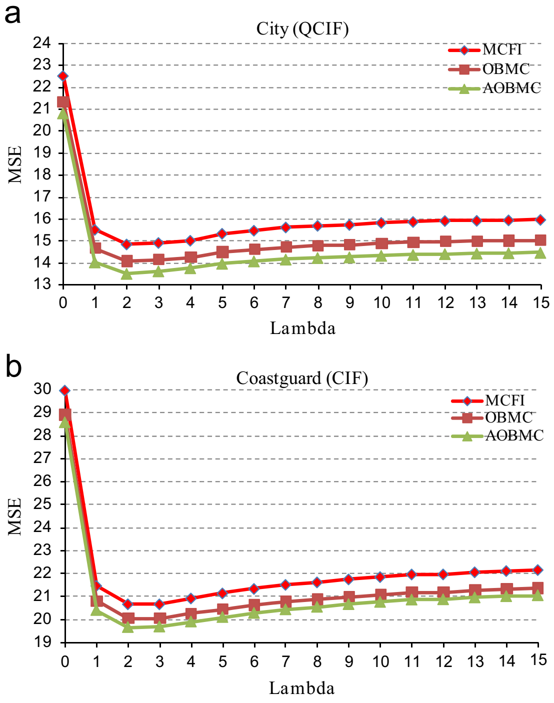

Fig. 5. Relationship between MSE and parameter λ over *City* (QCIF) and *Coastguard* (CIF). (a) Relationship between MSE and parameter λ over *City* (QCIF). (b) Relationship between MSE and parameter λ over *Coastguard* (CIF).

4.2. Peformance evaluation

In this subsection, five QCIF sequences and five CIF sequences are selected to evaluate the performance of the proposed frame rate reduction. The first 150 frames of each test sequence are down-sampled by the direct down-sampling, MCTF [16] and the proposed algorithm. The down-sampled sequences are then interpolated by MCFI [1], OBMC [11] and AOBMC [9]. In MCTF method, group of picture (GOP) is set to be two.

The average MSE of all the down-sampled and interpolated frames compared with the corresponding original ones is presented in Table 3. Here, 'Direct' means the direct down-sampling algorithm. It can be observed that the average MSE of MCTF is lower than that of direct down-sampling method. However, the MSE drop is rather small. On the contrary, the MSE of the proposed algorithm is much lower than those of the former two methods. Especially, for *City* (QCIF), the average MSE gap between the proposed method and the other two competing methods are more than 1.5 under MCFI method. And for *Coastguard* (CIF), the average MSE gaps are more than 1.8 compared with both Direct and MCTF methods. This is greatly attributed to the property that the proposed down-sampling algorithm is able to maintain much more information in the down-sampled sequences.

Fig. 6 provides the PSNR comparisons over each interpolated frame by MCFI under different down-sampled

Table 3
MSE comparions under different down-sampling and up-sampling combinations.

Resolution	Sequences	Average MSE of both down-sampled and up-sampled frames								
		MCFI			OBMC			AOBMC		
		Direct	MCTF	Proposed	Direct	MCTF	Proposed	Direct	MCTF	Proposed
QCIF	*City*	16.301	16.366	14.862	15.265	15.336	14.089	14.683	14.742	13.525
	Football	156.814	155.527	150.311	147.721	146.690	142.794	146.423	145.335	141.286
	News	13.018	13.062	12.452	12.458	12.468	12.012	12.484	12.485	11.969
	Salesman	1.915	1.867	1.708	1.764	1.719	1.612	1.730	1.684	1.568
	Hall	3.841	3.720	3.511	3.686	3.560	3.397	3.616	3.491	3.308
Avg		**38.378**	**38.108**	**36.568**	**36.179**	**35.955**	**34.781**	**35.787**	**35.547**	**34.331**
CIF	*Foreman*	12.074	12.030	11.185	11.487	11.446	10.725	11.435	11.388	10.614
	Mobile	52.576	52.093	49.178	44.983	44.524	42.195	39.781	39.393	37.308
	Stefan	85.925	85.411	81.832	81.218	80.770	77.602	80.495	80.029	76.738
	Flower	24.571	24.285	23.276	22.316	22.018	21.283	22.568	22.296	21.466
	Coastguard	22.699	22.481	20.672	21.810	21.581	20.050	21.491	21.266	19.647
Avg		**39.569**	**39.260**	**37.228**	**36.363**	**36.068**	**34.371**	**35.154**	**34.874**	**33.155**

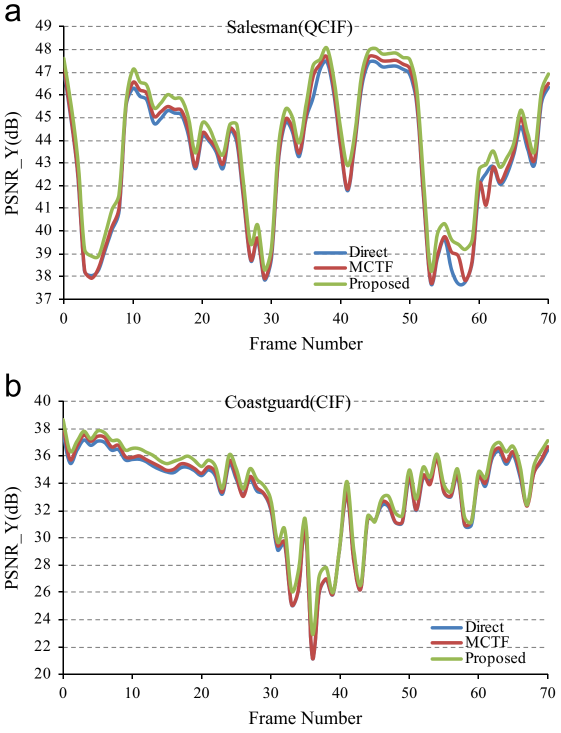

Fig. 6. PSNR comparisons over each interpolated frame with MCFI under different down-sampled sequences for *Salesman* (QCIF) and *Coastguard* (CIF). (a) PSNR of each interpolated frame for *Salesman* (QCIF) under different down-sampled sequences. (b) PSNR of each interpolated frame for *Coastguard* (CIF) under different down-sampled sequences.

sequences for *Salesman* (QCIF) and *Coastguard* (CIF). It can be seen that for the majority interpolated frames, MCTF achieves better result than direct down-sampling method. This is because MCTF is able to transfer some information of the up-sampled frame to the down-sampled frames via the filtering operation along the temporal direction. However, the performance improvement of interpolated frames is marginal. In contrast, the proposed down-sampling algorithm is able to absorb much more information, which is very important for MCFI, during down-sampling. Particularly, for *Salesman* (QCIF), the PSNR gains are nearly 0.8 dB from the 42th to the 50th interpolated frames. And for *Coastguard* (QCIF), the PSNR gains are nearly 1 dB from the 10th to the 20th interpolated frames.

Visual comparisons of interpolated frames over *Foreman* (CIF) and *Mobile* (CIF) are illustrated in Figs. 7 and 8, respectively. In Fig. 7, there are significant blocking artifacts exhibited around the nose in the MCFI results from the direct down-sampled and MCTF down-sampled sequences. However, these serious blocking artifacts are hardly perceived in the interpolated frame from the down-sampled sequence by the proposed algorithm. In the Figs. 8 and 11 cannot be clearly observed in the interpolated frames from the direct down-sampled and MCTF down-sampled sequences. Whereas, in the interpolated frame from the down-sampled sequence by the proposed method, 'Fig. 11' can be clearly observed. It is noted that in both Figs. 7 and 8, the up-sampled methods are the same, i.e., MCFI [1]. It is obvious that the proposed algorithm is able to absorb more information into the down-sampled sequences, and consequently generate interpolated frames with much higher visual quality.

4.3. Application in low bitrate video coding

To further demonstrate the desirable information preserving ability of the proposed algorithm, a frame rate reduction based low bit rate video coding is provided, which is depicted in Fig. 9. In the provided low bit rate video coding, the frame rate of input sequence is first reduced by half, and then the down-sampled sequence is encoded. At the decoder side, the decoded sequence is interpolated to the original frame rate via frame rate up conversion. However, the interpolated frames usually have lower PSNR values compared with the previous and following frames, which are directly encoded by H.264/AVC. For example, if the direct down-sampling is applied, there will exhibit quite non-smooth picture

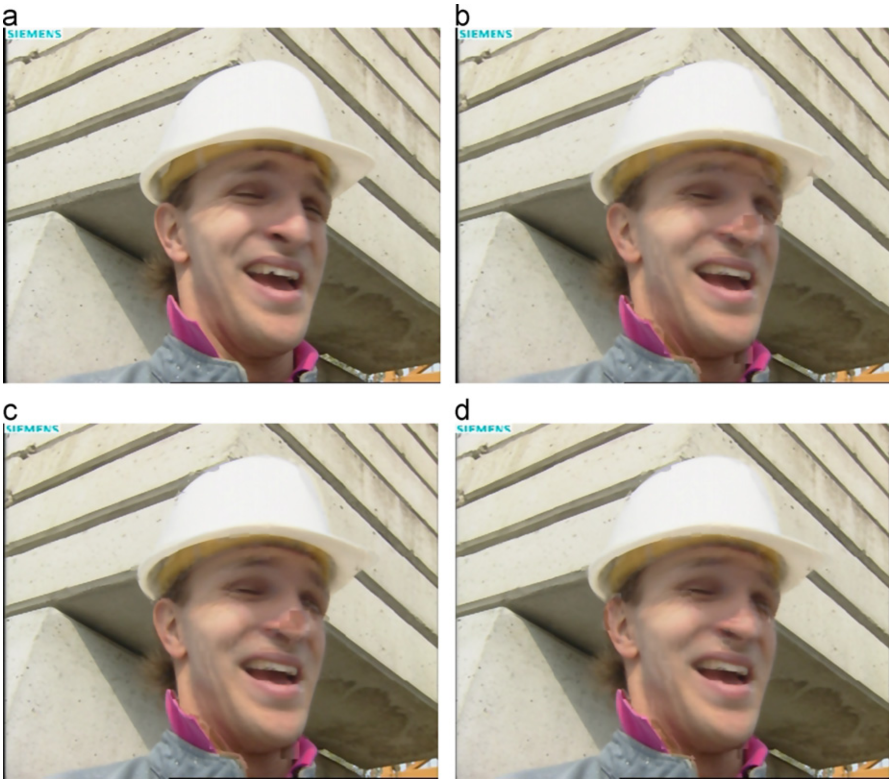

Fig. 7. Comparisons of the 6th interpolated frame in *Foreman* (CIF) from different down-sampled sequences. (a) Original (b) MCFI result from the direct down-sampled sequence (32.122 dB), (c) MCFI result from down-sampled sequence by MCTF (32.336 dB) (d) MCFI result from the down-sampled sequence by the proposed algorithm (33.235 dB).

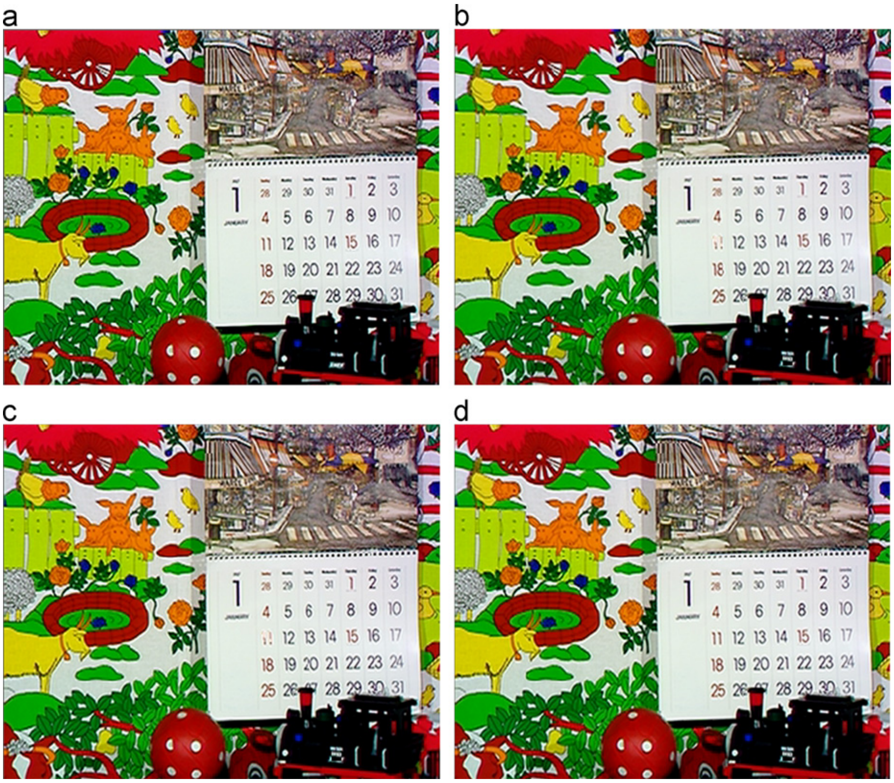

Fig. 8. Comparisons of the 25th interpolated frame in *Mobile* (CIF) from different down-sampled sequences. (a) Original (b) MCFI result from the direct down-sampled sequence (27.512 dB), (c) MCFI result from down-sampled sequence by MCTF (27.490 dB) (d) MCFI result from the down-sampled sequence by the proposed algorithm (27.870 dB).

quality in the combined sequence (composed of decoded frames and those interpolated by frame rate up conversion) at the decoder side. However, applying the proposed down-sampling algorithm, more information can be absorbed into the previous and following frames leading to little PSNR fluctuation between the encoded frames and interpolated frames.

In this subsection, *Salesman* (QCIF), *News* (QCIF), *Foreman* (CIF) and *Mobile* (CIF) are selected to carry out the provided low bit rate video coding. We first reduce the frame rate of the first 100 frames of each test sequence by half using Direct, MCTF and the proposed method. And then the downsampled sequence is encoded by the reference software JM18. The encoding structure is IPPP. The decoded sequence is then interpolated by OBMC [11].

The rate distortion curves for each test sequence at low bit rate video coding are depicted in Fig. 10, where Anchor represents encoding the input sequence without frame reduction, Direct, MCTF and Proposed represent reducing the frame rate by half prior to encoding, and then the decoded sequence is interpolated by OBMC [11]. Obviously, Direct, MCTF and the proposed method achieve better performance than Anchor does at low bit rate. Besides, the rate distortion performance of Direct, MCTF and the proposed method are very similar. However, the PSNR fluctuations between the interpolated frames and those encoded by H.264/AVC applying Direct and MCTF down-sampling methods are larger than those using the proposed method.

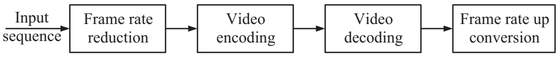

Fig. 9. Flowchart of frame rate reduction based low bitrate video coding.

Fig. 11 depicts the curves of consumed rate and PSNR variances for *Foreman* (CIF), *Mobile* (CIF), *News* (QCIF) and *Salesman* (QCIF). It can be observed that the interpolated frames in the proposed method have lower PSNR variances compared with those achieved by other down-sampling methods.

Figs. 12 and 13 provide the visual comparisons of the 7th interpolated frames for *Foreman* (CIF) and the 33rd frame for *Mobile* (CIF). It is noted that all the frames in Fig. 12(b–d) and Fig. 13(b–d) are interpolated by OBMC [11]. It can be observed that there exhibits serious blocking artifacts around the tongue region in the interpolated frame from down-sampled sequence by Direct and MCTF methods, whereas no blocking artifacts can be observed in the interpolated frame from the down-sampled sequence by the proposed method in Fig. 12. In Fig. 13(b), serious blocking artifacts around numbers “11”, “12”, “13” and “15” can be observed in the interpolated frame from down-sampled sequence by Direct method, and serious blocking artifacts around numbers “11”, “12”, “15” and “18” can be observed in the interpolated frame from down-sampled sequence by MCTF method. On the contrary, the blocking artifacts around those regions are greatly removed in the interpolated frame from down-sampled sequence by the proposed method. It is worth noticing that in Figs. 12 and 13, the bitrates of Direct, MCTF and the proposed method are the same. It further reveals that the proposed down-sampling method is of

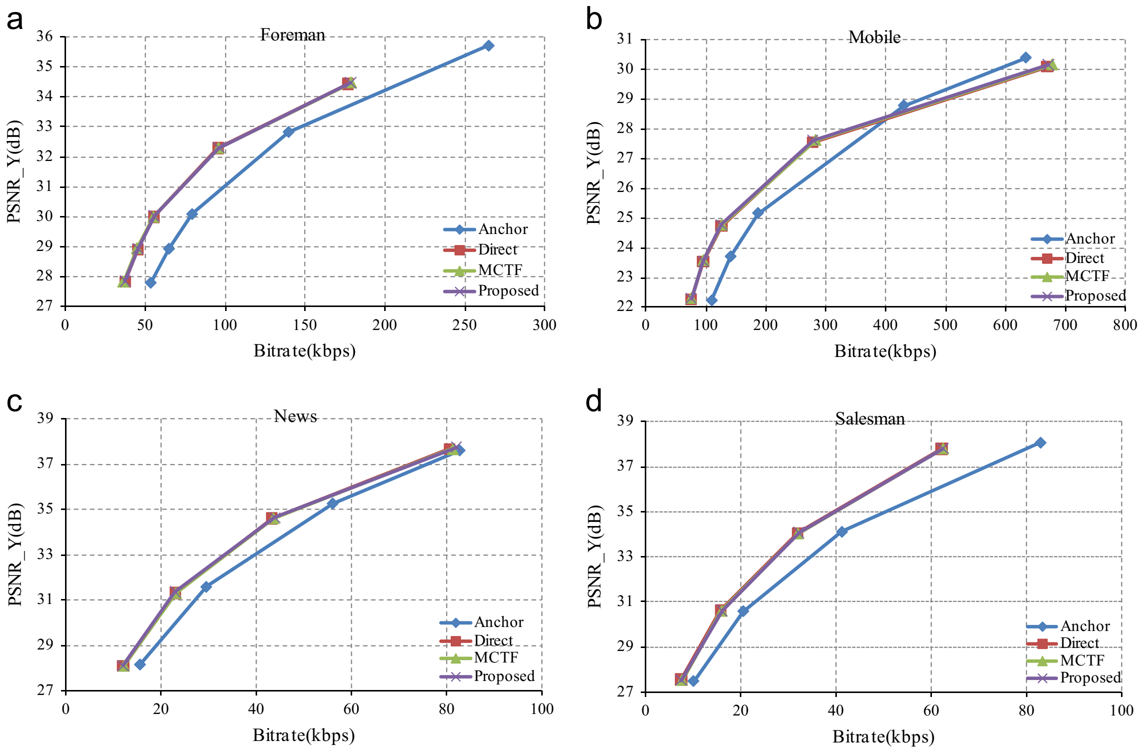

Fig. 10. Rate distortion curves for *Foreman* (CIF), *Mobile* (CIF), *News*(QCIF) and *Salesman*(QCIF).

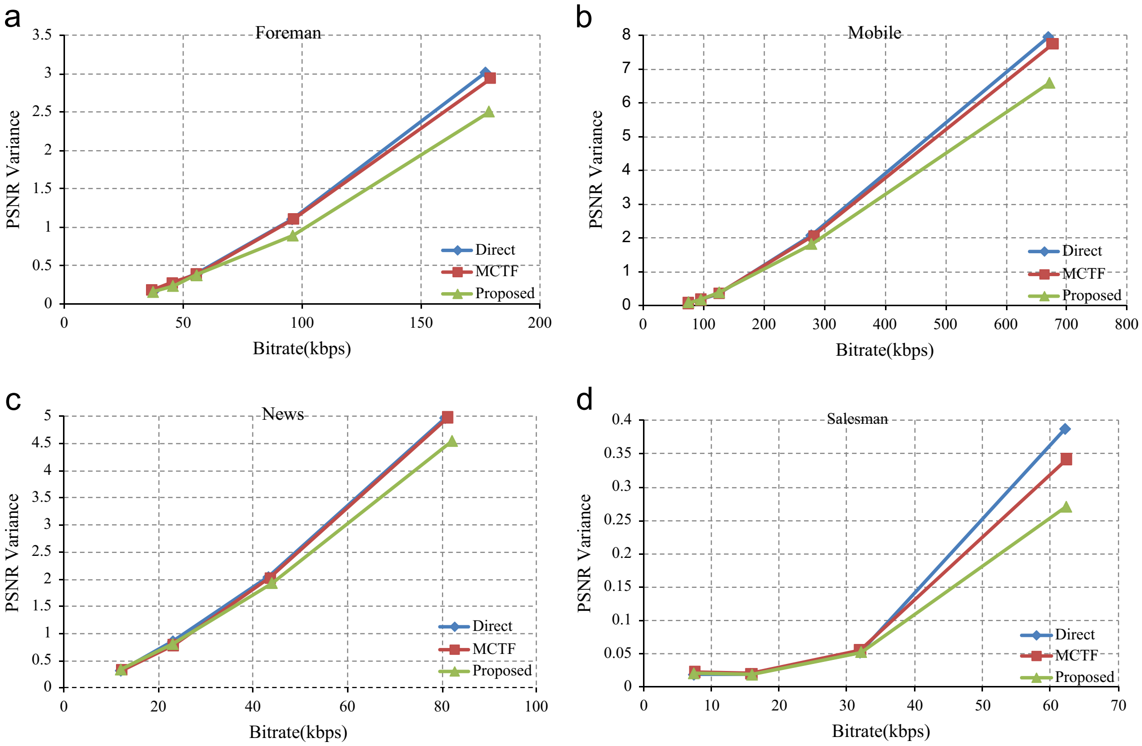

Fig. 11. Curves of Rate and PSNR variances for *Foreman* (CIF), *Mobile*(CIF), *News*(QCIF) and *Salesman*(QCIF).

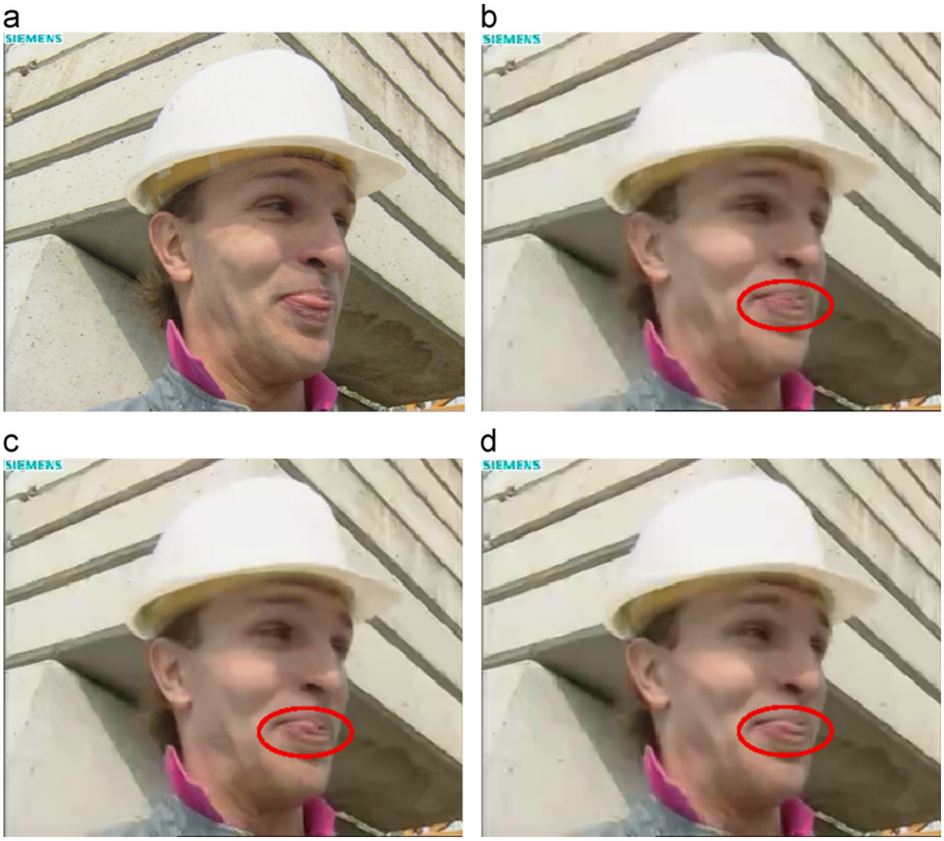

Fig. 12. Visual comparisons of the 7th frame under different coding methods for *Foreman* (CIF). (a) Original; (b) Direct with PSNR 32.617 dB (PSNR of previous and following frames are 33.147 dB and 33.201 dB); (c) MCTF with PSNR 32.583 dB (PSNR of previous and following frames are 33.124 dB and 33.169 dB); (d) Proposed with PSNR 32.874 dB (PSNR of previous and following frames are 33.199 dB and 33.240 dB).

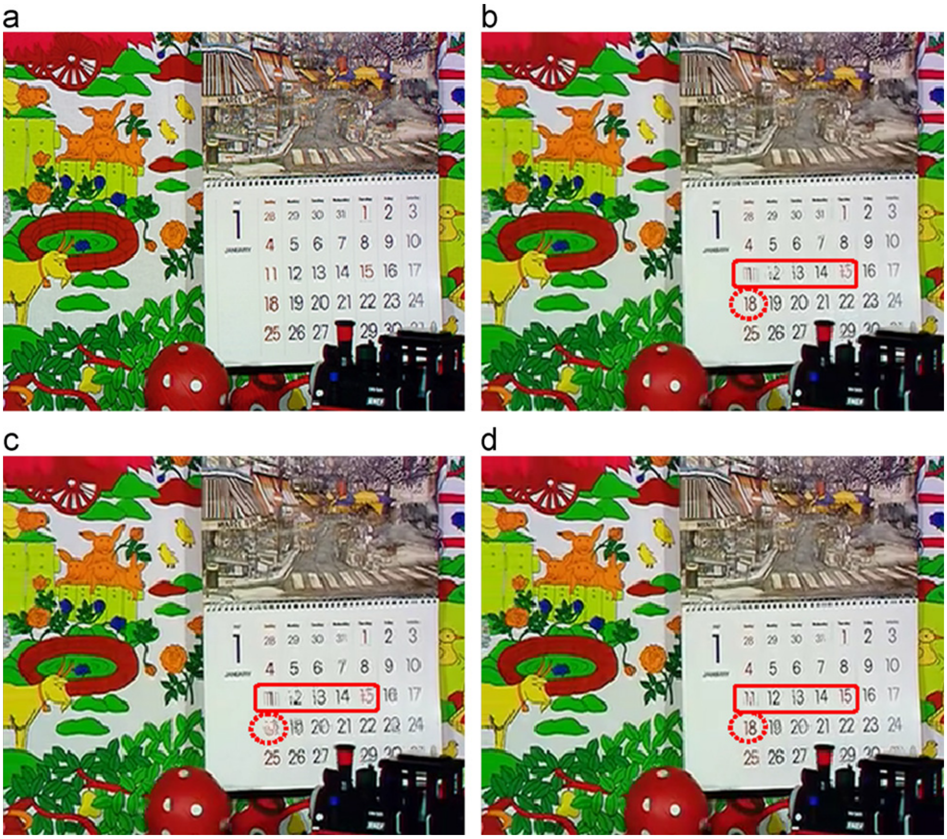

Fig. 13. Visual comparisons of the 33th frame under different coding methods for *Mobile* (CIF). (a) Original; (b) Direct with PSNR 25.755 dB (PSNR of previous and following frames are 32.821 dB and 32.838 dB); (c) MCTF with PSNR 25.234 dB (PSNR of previous and following frames are 32.855 dB and 32.865 dB); (d) Proposed with PSNR 25.623 dB (PSNR of previous and following frames are 32.764 dB and 32.806 dB).

Table 4
Average processing time (sec/frame) under different down-sampling algorithms.

Resolution	Sequences	Direct	MCTF	Proposed
QCIF	City	0.001	0.251	0.411
	Football	0.001	0.345	0.449
	News	0.001	0.243	0.391
	Salesman	0.001	0.229	0.424
	Hall	0.001	0.247	0.409
Avg		**0.001**	**0.263**	**0.417**
CIF	Foreman	0.008	1.015	1.725
	Mobile	0.008	0.987	1.583
	Stefan	0.008	1.096	1.635
	Flower	0.008	0.992	1.656
	Coastguard	0.008	1.029	1.498
Avg		**0.008**	**1.024**	**1.619**

desirable ability to remain more information in the down-sampled sequence.

4.4. Computational complexity analysis

The computational complexity of the proposed frame rate reduction algorithm mainly concentrates on the constructions of the interpolation matrix $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$, the constant vector $\mathbf{C}_q$, the matrix multiplication and matrix inverse operations involved in Eq. (19). It will take $\mathbf{O}(m^2n^2)$ to calculate $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$. The matrix multiplication and matrix inverse operation will take $\mathbf{O}(m^3n^3)$ and $\mathbf{O}(m^6n^6)$, respectively. It should be noted that since the majority elements of $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ and $\mathbf{C}_q$ are zeros, we can store $\mathbf{H}_{\vec{\mathbf{V}}_b}(q)$ and $\mathbf{C}_q$ using sparse matrix, which will further reduce the storage and computation complexities.

Table 4 provides the average processing time of each frame under different down-sampling algorithms on a typical computer (2.5 GHz Intel Dual Core, 4 GB Memory). It can be observed that Direct down-sampling algorithm has the smallest average processing time. This is because Direct is the most simple down-sampling method. The average processing time of MCTF is larger than Direct, since it involves motion estimation process. The proposed down-sampling algorithm has the largest average processing time, since it involves not only motion estimation, but also a series of matrix multiplication and matrix inverse operations. However, it is still acceptable with the appearance of more and more powerful computers.

5. Conclusions

An up-sampling oriented frame rate reduction algorithm is proposed in this paper. Different from traditional methods, where the frame rate reduction is independent from up-sampling, the proposed algorithm jointly considers the frame rate reduction and temporal up-sampling during down-sampling process. The proposed algorithm aims to generate a down-sampled frame that not only minimizes the difference between the original and down-sampled frame but also minimizes the difference between the original and the corresponding up-sampled frame. Consequently it is able to generate a down-sampled sequence, from which a higher quality up-sampled sequence can be interpolated. To facilitate the practical realization, a block-wise implementation of the proposed algorithm is also devised. Experimental results demonstrate that our algorithm can improve the quality of up-sampled frames compared with other down-sampling algorithms. Besides, the picture quality between interpolated frames and decoded frames can be smoothed by the proposed method under similar RD performance at low bit rate video coding.

Acknowledgement

This work was partially supported by National Science Foundation of China (61170195), the Joint Funds of National Science Foundation of China (U0935001&U1201255), the Upgrading Project of Shenzhen Key Laboratory (CXB201005260071A), and the Basic Research Plan in Shenzhen City (JC201105201110A).

References

[1] B.T. Choi, S.H. Lee, S.J. Ko, New frame rate up-conversion using bi-directional motion estimation, IEEE Transactions on Consumer Electronics 46 (3) (2000) 603–609.

[2] R. Castagno, P. Haavisto, G. Ramponi, A method for motion adaptive frame rate up-conversion, IEEE Transactions on Circuits and Systems for Video Technology 6 (5) (1996) 436–446.

[3] S.-H. Lee, Y.-C. Shin, S.-J. Yang, H.-H. Moon, R.-H. Park, Adaptive MC interpolation for frame rate up-conversion, IEEE Transactions on Consumer Electronics 48 (3) (2002) 444–450.

[4] G.I. Lee, B.W. Jeon R.H. Park, S.H. Lee, "Hierarchical motion compensated frame rate up-conversion based on the Gaussian/Laplacian pyramid," Proceedings IEEE International Conference Consumer Electronics, 2003, pp. 350–351.

[5] Z. Gan, L. Qi, X. Zhu, Motion compensated frame interpolation based on H.264 decoder, Electronics Letters 43 (2) (2007) 96–98.

[6] G. de Haan, P.W. Biezen, H. Huijgen, O.A. Ojo, True-motion estimation with 3-D recursive search block matching, IEEE Transactions on Circuits and Systems for Video Technology 3 (5) (1993) 368–379.

[7] T. Wiegand, G.J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 560–576.

[8] G. Dane, T. Nguyen, "Motion vector processing for frame rate up conversion", Proceedings IEEE International Conference Acoustics, Speech, and Signal Processing, ICASSP, 2004, pp. 309–312.

[9] 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 Transactions on Circuits and Systems for Video Technology 17 (4) (2007) 407–416.

[10] B. Girod, Efficiency analysis of multihypothesis motion-compensated prediction for video coding, IEEE Transactions on Image Processing 9 (2) (2000) 173–183.

[11] J. Zhai, K. Yu, J. Li, S. Li, A low complexity motion compensated frame interpolation method, Proceedings ISCAS 5 (2005) 4927–4930.

[12] Y. Zhang, D. Zhao, X. Ji, R. Wang, W. Gao, A spatio-temporal auto regressive model for frame rate up-conversion, IEEE Transactions on Circuits and Systems for Video Technology 19 (9) (2009) 1289–1301.

[13] Y. Zhang, D. Zhao, S. Ma, R. Wang, W. Gao, A motion aligned auto-regressive model for frame rate up conversion, IEEE Transactions on Image Processing 19 (5) (2010) 1248–1258.

[14] Y. Bandoh, S. Takamura, K. Kamikura, Y. Yashima, Adaptive down-sampling of frame rate for high frame rate video, Proceedings of the IEEE Picture Coding Symposium, PCS (2009) 129–132.

[15] Y. Bandoh, S. Takamura, K. Kamikura, Y. Yashima, "Temporal down-sampling algorithm of high frame rate video for reducing inter-frame prediction error," Proceedings IEEE International Conference Image Processing, ICIP, 2009, pp. 1013–1016.

[16] J. Ohm, Advances in scalable video coding, Proceedings of the IEEE 93 (1) (2005) 42–56.

[17] G. Dane, T.Q. Nguyen, Optimal temporal interpolation filter for motion-compensated frame rate up conversion, IEEE Transactions on Image Processing 15 (4) (2006) 978–991.