Joint Overlapped Block Motion Compensation Using Eight-Neighbor Block Motion Vectors for Frame Rate Up-Conversion

Ran Li, Minghu Wu, Zongliang Gan, Ziguan Cui, Xiuchang Zhu

Jisangsu Key Lab on Image Processing & Image Communication,

Nanjing University of Posts and Telecommunications, 210003, Nanjing, China

[e-mail: *liran358@163.com*]

*Corresponding author: Xiuchang Zhu

Received April 17, 2013; revised June 12, 2013; accepted September 22, 2013; published October 29, 2013

Abstract

The traditional block-based motion compensation methods in frame rate up-conversion (FRUC) only use a single uniquely motion vector field. However, there will always be some mistakes in the motion vector field whether the advanced motion estimation (ME) and motion vector analysis (MA) algorithms are performed or not. Once the motion vector field has many mistakes, the quality of the interpolated frame is severely affected. In order to solve the problem, this paper proposes a novel joint overlapped block motion compensation method (8J-OBMC) which adopts motion vectors of the interpolated block and its 8-neighbor blocks to jointly interpolate the target block. Since the smoothness of motion filed makes the motion vectors of 8-neighbor blocks around the interpolated block quite close to the true motion vector of the interpolated block, the proposed compensation algorithm has the better fault-tolerant capability than traditional ones. Besides, the annoying blocking artifacts can also be effectively suppressed by using overlapped blocks. Experimental results show that the proposed method is not only robust to motion vectors estimated wrongly, but also can to reduce blocking artifacts in comparison with existing popular compensation methods.

Keywords: Frame rate up-conversion (FRUC), 8-neighbor blocks, minimum mean square error (MMSE), Tikhonov regularization, overlapped block

A Letter version of this paper appeared in IEICE Transaction on Information and System, vol.E96-D, no.4, 2013. This version improves the previous algorithm and include a concrete analysis and supporting implementation results on the new algorithm. This work was supported by the Graduate Student Innovation Project of Jiangsu Province China (CXZZ12_0466 and CXZZ11_0390), National Natural Science Foundation of China (61071091 and 61271240), Natural Science Foundation of the Higher Education Institutions of Jiangsu Province China (12KJB510019), and Technology Research Program of Hubei Provincial Department of Education (D20121408).

http://dx.doi.org/10.3837/tiis.2013.10.007

1. Introduction

Frame rate up-conversion (FRUC) refers to the technique that constructs a high frame rate video by periodically inserting new frames into an input lower frame rate video. FRUC has wide applications in video processing, such as the conversion between two display formats with different frame rates, low bit-rate video communication, the reduction of motion blur of the liquid crystal display (LCD) TVs, and side information generation in distributed video coding (DVC), etc [1-4]. Various FRUC algorithms have been developed, and a simple approach is to combine the pixel values at the same spatial location without considering object motions, e.g., frame repetition or frame averaging. Although these algorithms provide acceptable visual quality in the absence of motions, they can produce motion jerkiness and ghosting artifacts of moving objects. In order to remove these negative effects, a more appropriate approach is to perform frame interpolation along the motion trajectories. This is commonly referred to as motion-compensated frame rate up-conversion (MC-FRUC).

MC-FRUC consists of two steps, motion estimation (ME) and motion compensated interpolation (MCI). ME is a process of calculating a motion vector which must represent the true motion vector of objects in the image sequence since all interpolation processes are controlled by the motion vectors. Most MC-FRUC algorithms utilize the block-matching algorithm (BMA) for ME on account of its simpleness and easy implementation [5-6]. The unidirectional ME is a common approach but introduces the overlapped (multi-passing of motion trajectories) and hole (no motion trajectory is passing) regions in the interpolated frame. To avoid holes and overlapped regions, [7] proposed bi-directional ME (BME) which arranges the unique motion vector for every block in the interpolated frame. However, motion vectors estimated using BME are often not faithful to true object motions, several approaches for obtaining more accurate BME has been proposed in some recent work [8-9]. The above-mentioned ME algorithms mostly use the full search so as to high computational complexity and low accuracy. In order to improve defects of full search, [10] proposed a 3-D recursive ME (3DRS) which has been applied to several MCI schemes. The ME may sometimes result in non-consistent motion fields, thus some motion vector post processing methods [11-12] were proposed to smooth the motion fields. MCI is a process to make an interpolated frame with motion vectors obtained from ME and usually performs block by block. Whereas block edges may not always be consistent with the heterogeneous object edges, and thus blocking artifacts are usually perceived in the regions where one block has a significantly different motion compared with its neighbors. By extending traditional MCI, overlapped block motion compensation (OBMC) [13-14] was employed for its efficiency of reducing blocking artifacts. However, OBMC may result in blurring or over-smoothing artifacts in case of non-consistent motion regions since it assigns fixed weights for neighboring blocks. For better adjusting the weights of OBMC, an adaptive OBMC (AOBMC) [15] was also proposed to tune the weights of different blocks according to the reliability of neighboring motion vectors. Recently, [16-17] blazed a new trail for FRUC by introducing auto-regressive (AR) model in which each pixel in the interpolated frame is approximated by a linear combination of the pixels in a square neighborhood in the reference frames. In addition, [18] brought in Bayesian concept to adaptively fuse multiple predictions interpolated by different ME strategies.

In this paper, we propose a motion compensation method 8J-OBMC for FRUC based on our previous work [19]. The proposed method cannot use a single motion vector to perform block-based MCI but adopts motion vectors of the interpolated block and its 8-neighbor blocks to jointly make a better prediction under the two assumptions that the temporal symmetry between previous and following frames and the smoothness of motion vector field. Different from 8J-MCI proposed by [19], we use the overlapped blocks to interpolate the current frame in order to suppress blocking artifacts. The results of simulation show that the proposed method is not only robust to motion vector estimated wrongly, but also can reduce blocking artifacts than 8J-MCI. Besides, the MC-FRUC algorithm comprised of BME and our joint compensation method also obtains the better performance in contrast with existing FRUC algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews the basic MC-FRUC model and discusses its existing problem. Details of the proposed method are presented in Section 3. Section 4 reports and discusses the simulation results of the proposed method followed by conclusions in Section 5.

2. The Basic MC-FRUC Model and Its Existing Problem

The basic MC-FRUC model can be described briefly using probability theory [18]. Suppose $\boldsymbol{f}_t$ is the interpolated frame to be estimated, and the previous and the following neighboring frames of $\boldsymbol{f}_t$ refer to $\boldsymbol{f}_{t-1}$ and $\boldsymbol{f}_{t+1}$, respectively. The goal of FRUC problem is to predict a pixel value with maximum probability for each pixel of $\boldsymbol{f}_t$ based on $\boldsymbol{f}_{t-1}$ and $\boldsymbol{f}_{t+1}$. The mathematical formulation of this problem is

$$\hat{\boldsymbol{f}}_t = \arg\max_{\boldsymbol{f}_t} \Pr(\boldsymbol{f}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}) \quad . \tag{1}$$

Here, Pr $(\cdot)$ is the probability density function.

Since the three consecutive frame $\boldsymbol{f}_t$, $\boldsymbol{f}_{t-1}$ and $\boldsymbol{f}_{t+1}$ should be consistent and form a continuous scene, there is motion to link the interpolated frame $\boldsymbol{f}_t$ with the reference frames $\boldsymbol{f}_{t-1}$ and $\boldsymbol{f}_{t+1}$. By utilizing motion field $\boldsymbol{m}_t$ of $\boldsymbol{f}_t$, (1) can be reformulated as

$$\begin{aligned}\hat{\boldsymbol{f}}_t &= \arg\max_{\boldsymbol{f}_t} \int \Pr(\boldsymbol{f}_t, \boldsymbol{m}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}) d\boldsymbol{m}_t \\ &= \arg\max_{\boldsymbol{f}_t} \int \Pr(\boldsymbol{f}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}, \boldsymbol{m}_t) \Pr(\boldsymbol{m}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}) d\boldsymbol{m}_t \end{aligned} \quad . \tag{2}$$

The intuitive interpretation of this formulation is as follows. Firstly, we get the probability distribution of motion field $\boldsymbol{m}_t$ of the interpolated frame $\boldsymbol{f}_t$. Second, the probability distribution of $\boldsymbol{f}_t$ is calculated by considering all possible motion fields. At last, we find a pixel value with maximum probability for each pixel of $\boldsymbol{f}_t$. However, the probability of motion field is difficult to model, so (2) cannot be solved directly. In practice, this model is commonly divided into two parts,

$$\hat{\boldsymbol{m}}_t = \arg\max_{\boldsymbol{m}_t} \Pr(\boldsymbol{m}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}) , \quad (3)$$

$$\hat{\boldsymbol{f}}_t = \arg\max_{\boldsymbol{f}_t} \Pr(\boldsymbol{f}_t \mid \boldsymbol{f}_{t-1}, \boldsymbol{f}_{t+1}, \boldsymbol{m}_t) . \quad (4)$$

The goal of (3) is to find the most probable motion field of $\boldsymbol{f}_t$, and the process is called ME in basic MC-FRUC model. Since mistakes can still happen in the estimated motion field $\hat{\boldsymbol{m}}_t$, the motion vector analysis (MA) is commonly performed after ME to correct some vectors. The model (4) is the mathematical formulation of MCI which estimates the interpolated frame based on the estimated motion field $\hat{\boldsymbol{m}}_t$.

In conclusion, the basic MC-FRUC model can be summed up as **Fig. 1**. Since the quality of interpolated frame estimated by MCI mainly depends on the accuracy of the single motion field $\hat{\boldsymbol{m}}_t$, the motion that links the interpolated frame $\boldsymbol{f}_t$ with the pair of reference frames $\boldsymbol{f}_{t-1}$ and $\boldsymbol{f}_{t+1}$ is the key factor to be consider in the basic MC-FRUC model. However, the true motion field cannot be obtained in practice, and it can only be approached by some advanced ME and MA algorithm. We can also see by (3) that the estimated motion field $\hat{\boldsymbol{m}}_t$ just appears with maximum probability. In other words, other motion fields might also come out with the smaller probability. This is the main problem in basic MC-FRUC model, and the source of the problem is that the two formulas (3) and (4) only consider the most probable motion field but ignore other motion fields. In order to solve the problem, this paper proposes a strategy that uses the multiple motion fields as much as possible to jointly compensate the interpolated frame. The proposed algorithm will be discussed in the following section.

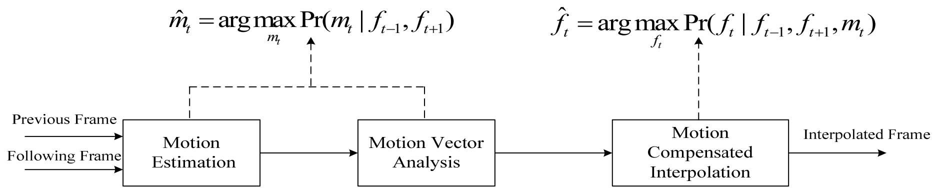


Fig. 1. The block diagram of basic MC-FRUC model

3. Description of the Proposed Method

In the basic MC-FRUC model, once the estimated motion field has many incorrect motion vectors, the quality of interpolated frame will severely be affected. A straightforward method to solve this problem is to consider all possible motion fields as (2). It is difficult to model the probability of motion field Pr ($\boldsymbol{m}_t|\boldsymbol{f}_{t-1},\boldsymbol{f}_{t+1}$), but the next best thing is to choose several motion fields with high probability to jointly make a prediction of the interpolated frame. In order to avoid high complexity, the MCI will be performed block by block. As shown in **Fig.2**, the proposed algorithm consists of two parts: the first part is to construct a candidate motion vector set of the interpolated block, and the second part is to utilize the multiple motion vectors from the candidate motion vector set to jointly interpolate the target block.

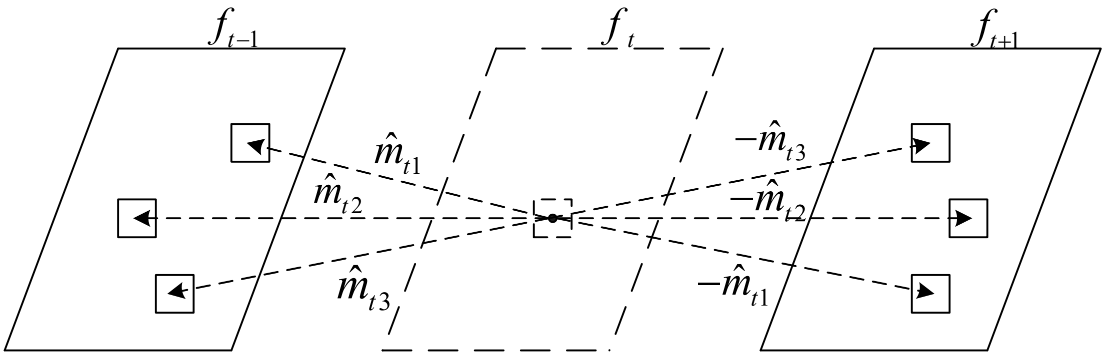
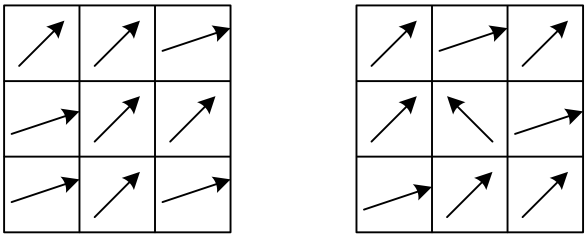


Fig. 2. Multiple motion vectors jointly interpolate the target block

3.1 Construction of Candidate Motion Vectors

The candidate motion vectors of the interpolated block should be close to its true motion vector. Since motion vector fields are usually smooth, motion vectors from neighboring blocks are commonly closer to the true vector of the current block. Most motion vector analysis algorithms use the assumption to correct outliers appearing in motion vector field, as shown in **Fig. 3**. Likewise, depending on the assumption, the candidate motion vector set of the interpolated block can be constructed by motion vectors of the interpolated block and its 8-neighbor blocks. There will always be some motion vectors estimated correctly or even true motion vector in the candidate motion vector set, so this method has the better fault-tolerant capability than traditional ones using a single motion vector with maximum probability.



(a) Smooth region **(b)** Outlier

Fig. 3. Motion vector field models

In this paper, we perform bi-directional ME (BME) algorithm proposed in [20] which briefly described as follows. First of all, a block matching algorithm using full search is used to estimate the motion field between the previous frame $\boldsymbol{f}_{t-1}$ and the following frame $\boldsymbol{f}_{t+1}$. For each block $\boldsymbol{x}$ in the previous frame $\boldsymbol{f}_{t-1}$, its motion vector directing at the block in interpolated frame $\boldsymbol{f}_t$ is derived as half the motion vector directing at the matching block in the following frame $\boldsymbol{f}_{t+1}$. However, this rigid block-based ME scheme fails to capture all aspects of the motion field, and if frame interpolation is performed, overlapped and uncover areas will appear. This is because the motion vectors obtained do not necessarily go through the center of each non-overlapped block in the interpolated frame. In order to make each non-overlapped block in the interpolated frame owns a single motion vector, the motion vector nearest to the center of the interpolated block is selected from the available candidate vectors obtained in the previous step. Then, the BME refinement is performed in search range confined to a small displacement around the initial block position, and the block matching algorithm uses the sum of bilateral absolute differences (SBAD) [15] based on the temporal symmetry between previous and following frames.

3.2 Proposed Joint Compensation Method

Suppose $\boldsymbol{x}_{tp}$ is the block interpolated using backward candidate motion vectors $\boldsymbol{m}_{ti}$, $i = 1,2,..,9$, and $\boldsymbol{x}_{tf}$ is the block interpolated using forward candidate motion vectors $-\boldsymbol{m}_{ti}$, $i = 1,2,..,9$, under the assumption of temporal symmetry. Note that $\boldsymbol{x}_{tp}$ and $\boldsymbol{x}_{tf}$ are the vectorized signal through raster scanning. The candidate matching blocks $\boldsymbol{\phi}_{ti}$ in the previous frame $\boldsymbol{f}_{t-1}$ can be found by using the following formula,

$$\phi_{ti}(\boldsymbol{s}) = f_{t-1}(\boldsymbol{s} + \boldsymbol{m}_{ti}) \quad i = 1,2,...,9 \quad . \tag{5}$$

Here, $\boldsymbol{s}$ denotes a pixel position in the interpolated block. Likewise, the candidate matching blocks $\boldsymbol{\psi}_{ti}$ in the following frame $\boldsymbol{f}_{t+1}$ can also be found by using the following formula,

$$\psi_{ti}(\boldsymbol{s}) = f_{t+1}(\boldsymbol{s} - \boldsymbol{m}_{ti}) \quad i = 1,2,...,9 \, . \tag{6}$$

Then, the $\boldsymbol{x}_{tp}$ can be predicted by the linear weighted sum of all candidate matching blocks $\boldsymbol{\phi}_{ti}$ in the previous frame $\boldsymbol{f}_{t-1}$, that is,

$$\boldsymbol{x}_{\text{tp}} = \sum_{i=1}^{9} \alpha_{\text{i}} \boldsymbol{\phi}_{\text{ti}} + \boldsymbol{n}_1 = \boldsymbol{\Phi\alpha} + \boldsymbol{n}_1$$

(7)

Here, the coefficients α_{i} , which compose the column vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2,\ldots, \alpha_9]^{\text{T}}$, determine the weight of each candidate matching blocks, and the dictionary formed by the candidate matching blocks is $\boldsymbol{\Phi} = [\ \boldsymbol{\phi}_{\text{t1}},\ \ \boldsymbol{\phi}_{\text{t2}},\ldots,\ \ \boldsymbol{\phi}_{\text{t9}}]$. Assuming the noise component $n_{1\text{k}}$ from $\boldsymbol{n}_1 = [n_{11},n_{12},...,n_{1\text{L}}]^{\text{T}}$ has been generated independently, where L is the length of $\boldsymbol{x}_{\text{tp}}$, and $n_{1\text{k}}$ is chosen to be a Gaussian distribution with mean zero and variance $\sigma_1^{\,2}$, that is, $\Pr(n_{1\text{k}}) = \text{N}(0, \sigma_1^{\,2})$. Similarly, the $\boldsymbol{x}_{\text{tf}}$ can be predicted by

$$\boldsymbol{x}_{\text{tf}} = \sum_{i=1}^{9} \beta_{\text{i}} \boldsymbol{\psi}_{\text{ti}} + \boldsymbol{n}_2 = \boldsymbol{\Psi\beta} + \boldsymbol{n}_2 \quad .$$

(8)

Here, $\Pr\,(n_{2\text{k}}) = \text{N}(0, \sigma_2^{\,2})$. The probability distribution of residual error $\boldsymbol{e}$ can be calculated by (7) and (8),

$$\Pr(\boldsymbol{e} = \boldsymbol{x}_{\text{tp}} - \boldsymbol{x}_{\text{tf}}) = \frac{1}{[2\pi(\sigma_1^2 + \sigma_2^2)]^{L/2}} \exp\{-\frac{\| \boldsymbol{e} - (\boldsymbol{\Phi\alpha} - \boldsymbol{\Psi\beta}) \|_2^2}{2(\sigma_1^2 + \sigma_2^2)}\} \quad .$$

(9)

The mean square error is that

$$E(\| \boldsymbol{e} \|_2^2) = (\sigma_1^2 + \sigma_2^2) \cdot L + \| \boldsymbol{\Phi\alpha} - \boldsymbol{\Psi\beta} \|_2^2 \quad .$$

(10)

On the basis of the assumption that the temporal symmetry between previous and following frames, the $\boldsymbol{x}_{\text{tp}}$ should be similar to the $\boldsymbol{x}_{\text{tf}}$.. Therefore, the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be computed by the minimum mean square error (MMSE) criterion. It is noted that the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ cannot be zero vector since one candidate block at least contributes to the prediction of the interpolated block, so we need to add constrains on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Above all, the optimal model is as follows,

$$\{\boldsymbol{\alpha}, \boldsymbol{\beta}\} = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \| \boldsymbol{\Phi\alpha} - \boldsymbol{\Psi\beta} \|_2^2,$$

$$s.t.\ \ \boldsymbol{u}^{\text{T}}\boldsymbol{\alpha} = 1,\ \boldsymbol{u}^{\text{T}}\boldsymbol{\beta} = 1 \quad .$$

(11)

Here, $\boldsymbol{u}$ is a full-ones vector.

However, without prior knowledge of the ‘truth’, the model (11) often produces over-fitting. To reducing the bad effects caused by over-fitting, the most common approach is to regularize the MMSE model using Tikhonov regularization which imposes an L2 penalty on the norm of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, that is,

$$\{\boldsymbol{\alpha},\boldsymbol{\beta}\} = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}}\{\|\boldsymbol{\Phi}\boldsymbol{\alpha}-\boldsymbol{\Psi}\boldsymbol{\beta}\|_2^2 + \lambda(\|\boldsymbol{\Gamma}_\alpha\boldsymbol{\alpha}\|_2^2 + \|\boldsymbol{\Gamma}_\beta\boldsymbol{\beta}\|_2^2)\},$$

$$s.t.\ \ \boldsymbol{u}^{\mathrm{T}}\boldsymbol{\alpha}=1, \boldsymbol{u}^{\mathrm{T}}\boldsymbol{\beta}=1 \quad , \tag{12}$$

where $\boldsymbol{\Gamma}_\alpha$ and $\boldsymbol{\Gamma}_\beta$ are known as the Tikhonov matrix [21]. The $\boldsymbol{\Gamma}_\alpha$ and $\boldsymbol{\Gamma}_\beta$ terms allow the imposition of prior knowledge on the solution $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. In our case, we can exploit the approach that the candidate matching blocks using the motion vectors closer to the true motion vectors should be given larger weight than the candidate matching blocks using the motion vectors far from the true motion vectors. It is obvious that if the motion vector is closer to the true motion vector, the candidate block in the previous frame found by it, is more similar to the corresponding candidate block in the following frame on account of temporal symmetry. Therefore, we proposed the diagonal $\boldsymbol{\Gamma}_\alpha$ and $\boldsymbol{\Gamma}_\beta$ in the form of

$$\boldsymbol{\Gamma}_\alpha = \boldsymbol{\Gamma}_\beta = \begin{bmatrix} \|\boldsymbol{\phi}_{t1}-\boldsymbol{\psi}_{t1}\|_2^2 & & & \\ & \|\boldsymbol{\phi}_{t2}-\boldsymbol{\psi}_{t2}\|_2^2 & & \\ & & \ddots & \\ & & & \|\boldsymbol{\phi}_{t9}-\boldsymbol{\psi}_{t9}\|_2^2 \end{bmatrix} . \tag{13}$$

With this structure, $\boldsymbol{\Gamma}_\alpha$ and $\boldsymbol{\Gamma}_\beta$ penalize weights of large magnitude assigned to the candidate blocks using the motion vectors which are far from the true motion vectors. In order to solve the model (12), a new model equivalent to model (12) is generated as follows:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}}\{\|\boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{\Gamma}\boldsymbol{w}\|_2^2\},$$

$$s.t.\ \ \boldsymbol{p}^{\mathrm{T}}\boldsymbol{w}=1,\ \ \boldsymbol{q}^{\mathrm{T}}\boldsymbol{w}=1, \tag{14}$$

where $\boldsymbol{X} = [\boldsymbol{\Phi},-\boldsymbol{\Psi}]$, $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\Gamma}_\alpha, \boldsymbol{\Gamma}_\beta)$, $\boldsymbol{w} = [\boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\beta}^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{p} = [\boldsymbol{u}^{\mathrm{T}},\boldsymbol{0}^{\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{q} = [\boldsymbol{0}^{\mathrm{T}},\boldsymbol{u}^{\mathrm{T}}]^{\mathrm{T}}$. Then, $\hat{\boldsymbol{w}}$ can be calculated directly by the usual Tikhonov solution,

$$\hat{\boldsymbol{w}} = -\frac{1}{2}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}(\mu_1\boldsymbol{p}+\mu_2\boldsymbol{q}) \quad , \tag{15}$$

$$\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix} = \begin{bmatrix} \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{p} & \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{q} \\ \boldsymbol{q}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{p} & \boldsymbol{q}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{q} \end{bmatrix}^{-1} \begin{bmatrix}-2\\ -2\end{bmatrix} . \tag{16}$$

Here, λ is a scale factor that controls the relative effect of the Tikhonov-regularization term in the optimization (14). We found in practice that a value $\lambda \in [0.1,0.3]$ provided the best results, consequently, we use $\lambda = 0.25$ from point on.

Given the solution $\hat{\boldsymbol{w}}$ of model (14), we can use all candidate blocks to jointly interpolate the target block as follows,

$$\boldsymbol{x}_{\mathrm{t}} = \frac{1}{2}(\boldsymbol{\Phi}\boldsymbol{\alpha} + \boldsymbol{\Psi}\boldsymbol{\beta}) = \frac{1}{2}[\boldsymbol{\Phi}, \boldsymbol{\Psi}]\hat{\boldsymbol{w}} \quad . \tag{17}$$

However, if weight coefficients are computed by using non-overlapped blocks, the interpolation must be achieved by the linear weighted sum of these non-overlapped blocks. Therefore, the blocking artifacts can also appear since the non-overlapped block cannot guarantee to reasonably estimate the pixel value in edge regions where object occlusion leads to the probability that a block contains multiple motion vectors. In order to overcome this problem, the overlapped block can be introduced to perform the above-mentioned compensation scheme. Suppose the size of the interpolated block is $b \times b$, we enlarge its block size to $2b \times 2b$, as shown in **Fig. 4**. By using the motion vector of the interpolated block, we can find candidate overlapped blocks and obtain the result of interpolation by the linear weighted sum of them. However, the four region A, B,C and D in each interpolated block overlap the neighboring blocks, e.g. the region A overlaps the top left four neighboring blocks V1, V2,V3 and V4. Therefore, each pixel in the interpolated block has the four candidate estimates, and we get the final pixel value by using raised cosine window proposed by [13]. For the sake of simpler notations, we assume that $\boldsymbol{s}$ is located on the A region of the interpolated block $\boldsymbol{B}_{\mathrm{t}}$ which is the overlapped block corresponding to the vector $\boldsymbol{x}_{\mathrm{t}}$. Then, each pixel $\boldsymbol{B}_{\mathrm{t}}(\boldsymbol{s})$ is predicted using four candidate estimates from neighboring blocks V1, V2,V3 and V4. Let $\boldsymbol{B}_{\mathrm{vi}}(\boldsymbol{s})$ denotes the pixel value from the overlapped block $\mathrm{V_i}$ and $win_{\mathrm{i}}(\boldsymbol{s})$ denotes the corresponding weighting coefficient. So, the $\boldsymbol{B}_{\mathrm{t}}(\boldsymbol{s})$ can be computed by using the following formula,

$$\boldsymbol{B}_{\mathrm{t}}(\boldsymbol{s}) = \sum_{i=1}^{4} win_{\mathrm{i}}(\boldsymbol{s}) \cdot \boldsymbol{B}_{\mathrm{vi}}(\boldsymbol{s}) \, . \tag{18}$$

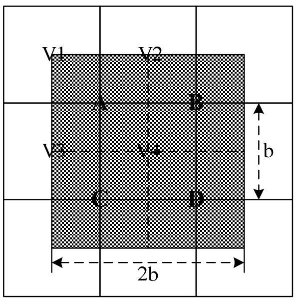


Fig. 4. Illustration of overlapped block

4. Experimental Results

The performance of the proposed joint compensation method 8J-OBMC for FRUC algorithm has been evaluated using 4 test sequences, which are in the standard CIF format and 30Hz frame rate. They are *Mobile* (containing rich details and slow motions), *Bus* (containing complex textures and medium speed motions), *Football* (containing fast motions) and *Foreman* (containing rich motions in the foreground and slight wobbles in the background). Every even frame of the first 100 frames in each test sequence is dropped and interpolated by the proposed algorithm. In all experiments, the block size is set 16×16 , and the motion vector field will be estimated using BME algorithm in which the radius of the search range in forward ME is 16 pixels and the radius of the search range in BME refinement is 2 pixels.

4.1 Evaluation of the Robustness to Incorrect motion vectors

In order to evaluate the fault-tolerant capability for motion vectors, the 8J-OBMC is compared with the OBMC [13] using popular MA algorithms in [7, 20, 22]. The average peak signal-to-noise ratios (PSNRs) of the 50 interpolated frames within each test sequence are presented in **Table 1**. Note that all methods use the motion vector field estimated by BME. It can be observed that the performance of our algorithm is a little better than the OBMC using MA algorithms in addition to the *Mobile* sequence. Since the *Mobile* sequence containing simple and slow motions has a flat motion field, the motion vector of the interpolated block is quite approach to the motion vectors of its 8-nerghbor blocks so that the interpolated block is less affected by its surrounding motion vectors. Therefore, for the *Mobile* sequence, the performance of our method is basically close to the compensation method using the median vector filter [22] or weighted median vector filter [20]. In conclusion, we can see that the 8J-OBMC can interpolate the target block while correcting the incorrect motion vector. However, the matrix operations in the proposed algorithm also increases computational burden which can be regarded as the cost of mixing in MA operating.

Table 1. The evaluation of fault-tolerant capability for the proposed compensation algorithm

	Mobile	*Bus*	*Football*	*Foreman*	Average
OBMC	27.02	26.61	22.42	34.23	27.57
OBMC + MA in [7]	27.17	27.24	22.65	34.56	27.91
OBMC + MA in [22]	29.13	25.86	22.11	33.47	27.69
OBMC + MA in [20]	**29.27**	27.42	22.59	34.66	28.49
8J-OBMC	29.03	**27.93**	**22.96**	**34.72**	**28.66**

4.2 Subjective Evaluation

We compare the subjective visual quality of the proposed 8J-OBMC with those of the traditional compensation methods MCI, OBMC, AOBMC [15] and 8J-MCI [19] which use the same motion vector field estimated by BME as the proposed method. It can be obviously seen from **Fig. 5** that the 13^{th} frame containing medium complex motions in *Foreman* recovered by MCI has some blocking artifacts (highlighted in red circle). Both OBMC and AOBMC suppress blocking artifacts in certain degree but result in blurring or over-smoothing artifacts. The frame recovered using 8J-MCI has the higher PSNR than OBMC and AOBMC, but the blocking artifacts are only slightly smoothed because of using non-overlapped blocks. However, the 8J-OBMC using overlapped blocks not only removes blocking artifacts but also has not over-smoothing artifacts. For the 12th frame containing slow motions in *Mobile*, as shown in **Fig. 6**, the numbers on calendar recovered by MCI, OBMC, AOBMC and 8J-MCI contain many annoying artifacts. On the other hand, the proposed scheme 8J-OBMC recovers these numbers visually pleasantly (highlighted in red circle). In additions, our method provides up to 0.47dB – 2.57dB better PSNR performance than other conventional ones.

(a) Original **(b)** MCI PSNR = 30.85 dB **(c)** OBMC PSNR = 31.65 dB

(d) AOBMC PSNR = 31.70 dB **(e)** 8J-MCI PSNR = 32.60 dB **(f)** 8J-OBMC PSNR = 32.86 dB

Fig. 5. The comparison of subjective visual quality for *Foreman* (13th frame)

(a) Original **(b)** MCI PSNR = 24.71 dB **(c)** OBMC PSNR = 25.32 dB

(d) AOBMC

PSNR = 25.30 dB

(c) 8J-MCI

PSNR = 26.81 dB

(d) 8J-OBMC

PSNR = 27.28 dB

Fig. 6. The comparison of subjective visual quality for *Mobile* (12th frame)

4.3 Objective Evaluation

In this section, three FRUC algorithms are selected as benchmarks, including the well-known 3DRS [10], MAAR (one of the state-of-the-art FRUC schemes) [17], DualME [8] and 8J-MCI with BME [19]. In the experiments, the motion search range for MAAR and DualME is set to 17×17. The ME block size used in these three benchmarks is 8×8. In order to compare with them, the MC-FRUC is constituted by BME algorithm and our 8J-OBMC. Its parameter settings are the same as the two experiments described above. The average PSNRs of the 50 interpolated frames are shown for each test sequence in **Table 2**. It can be observed that our MC-FRUC algorithm have the higher PSNR than other algorithms for all sequences except *Foreman*. The proposed scheme shows its superiority on sequences containing slow and medium speed motions, such as *Mobile* and *Bus*. For *Football* sequence containing fast motions, our method is also a little better than MAAR and 8J-MCI. However, for *Foreman* sequence, the best algorithm MAAR outperforms our method about 0.5 dB. The main reason is that *Foreman* sequence contains slight wobbles in the background, but our method does not take the global motion in account, so the quality of the interpolated frame decays in a certain degree. The average PSNRs of different algorithms are also presented in Table 2. It can be seen that the proposed method obtains the highest PSNR among all algorithms. In addition, the proposed method has a moderate computation complexity and the average processing time of each frame is about 5 s on a laptop (2.20 GHz Intel Core Duo CPU, 2 GB memory and MATLAB 7.6 simulation software).

Table 2. The PSNR (dB) comparison of different FRUC methods for CIF video sequences

	Mobile	*Bus*	*Football*	*Foreman*	Average
3DRS	27.03	25.99	22.28	33.51	27.20
MAAR	28.18	27.00	22.81	**35.28**	28.32
DualME	22.45	22.96	20.90	31.65	24.49
8J-MCI	28.67	28.02	22.96	34.81	28.62
8J-OBMC	**29.03**	**28.93**	**23.05**	34.72	**28.93**

5 Conclusions

In this paper, we presented a novel motion compensation method 8J-OBMC for FRUC. The proposed algorithm cannot use a single motion vector to perform block-based MCI but adopts motion vectors of the interpolated block and its 8-neighbor blocks to jointly interpolate the target block. Firstly, we perform the BME algorithm to compute the motion vectors of interpolated frame. Then, depending on the two assumptions that the temporal symmetry between previous and following frames and the smoothness of motion vector field, an MMSE model using Tikhonov regularization is used to compensate the interpolated frame. In order to suppress the annoying block artifacts, the overlapped block is applied to the interpolating process. The main advantage of our joint compensation approach is that it can interpolate the target block while correcting the mistakes exiting in motion field. Experimental results show that the proposed method is not only robust to motion vector estimated wrongly, but also can to reduce blocking artifacts by using overlapped block in comparison with existing popular compensation methods. In addition, the MC-FRUC algorithm comprised of BME and our joint compensation method outperforms MAAR (one of the state-of-the-art FRUC schemes) in the average PSNR for the test image sequence.

As future work, it is planned to enhance the performance of our joint compensation algorithm by the following two points:

(1) Construction of candidate motion vectors. In this paper, we use the motion vectors of 8-neighbor blocks around the target block to construct the candidate motion vector set. However, other methods to find candidate motion vectors still exist, and we will discuss some effective ones among them in the next step.

(2) MMSE model. The Tikhonov regularization in the proposed MMSE model is described by directly using the residual energy between the two candidate blocks. In future, the features extracted from candidate blocks will be designed to form the Tikhonov regularization so as to improve the accuracy of joint motion compensation.

References

[1] K. Hilman, Y. W. Park and Y. Kim. “Using motion compensated frame rate conversion for the correction of 3:2 pulldown artifacts in video sequence,” *IEEE Transaction on Circuits and Systems for Video Technology*, vol.10, no.6, pp.869-877, 2000. Article (CrossRef Link)

[2] Z. L. Gan and X. C. Zhu. “Study of frame-rate up conversion based on H.264,” *The Journal of China Universities of Posts and Telecommunications*, vol.14, no.1, pp.106-110, 2007. Article (CrossRef Link)

[3] J. Hwang, Y. Choi and Y. Choe. “Frame rate up-conversion technique using hardware-efficient motion estimator architecture for motion blur reduction for TFT-LCD,” *IEICE Transaction on Electronics*, vol.E94-C, no.5, pp.896-904, 2011. Article (CrossRef Link)

[4] B. Girod, A. Aaron and S. Rane, et al.. "Distributed video coding," *Processing of the IEEE*, vol.93, no.1, pp.71-83, 2005. Article (CrossRef Link)

[5] Y. K. Chen and S. Y. Kung. "Rate optimization by true motion estimation." in Proc. of *IEEE Workshop Multimedia Signal Process*, pp.187-194, 23-25 June, 1997. Article (CrossRef Link)

[6] C. C. Cheng, W. L. Hwang and Z. Shen, et al.. "Advanced motion compensation techniques for blocking artifacts reduction in 3-D video coding systems," in *Proc. of IEEE International Conference on Image Processing*, pp.89-92, 11-14 September, 2005. Article (CrossRef Link)

[7] B. T. Choi, S. H. Lee and S. J. Ko. "New frame rate up-conversion using bi-directional motion estimation," *IEEE Transaction on Consumer and Electron*, vol.46, no.3, pp. 603-609, 2000. Article (CrossRef Link)

[8] S. J. Kang, S. J. Yoo and Y. H. Kim. "Dual motion estimation for frame rate up-conversion," *IEEE Transaction on Circuits System and Video Technology*, vol.20, no.12, pp.1909-1914, 2010. Article (CrossRef Link)

[9] D. M. Wang, Zhang L and A. Vincent. "Motion-compensated frame rate up-conversion, part I: Fast multi-frame motion estimation," *IEEE Transaction on Broadcast*, vol.56, no.2, pp.133-141, 2010. Article (CrossRef Link)

[10] G. D. Haan, P. W. A. C. Biezen and H. Huijgen, et al.. "True motion estimation with 3-D recursive search block matching," *IEEE Transaction on Circuits System and Video Technology*, vol.3, no.5, pp. 368-379, 1993. Article (CrossRef Link)

[11] L. Alparone, M. Barni and F. Bartolini, et al.. "Adaptively weighted vector-median filters for motion fields smoothing," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2267-2270, May, 1996. Article (CrossRef Link)

[12] G. Dane and T. Nguyen. "Motion vector processing for frame rate up conversion," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.309-312, 17-21 May, 2004. Article (CrossRef Link)

[13] M. T. Orchard and C. J. Sullivan. "Overlapped block motion compensation: an estimation-theoretic approach," *IEEE Transaction on Image Processing*, vol. 3, no.9, pp. 693-699, 1994. Article (CrossRef Link)

[14] Su K J and Mersereau R M. "Motion estimation methods for overlapped block motion compensation," *IEEE Transaction on Image Processing*, vol.9, no.1, pp. 1509-1521, 2000. Article (CrossRef Link)

[15] B. D. Choi, J. W. Han and C. S. Kim, et al.. "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Transaction on Circuits System and Video Technology*, vol.17. no.7, pp. 407-416, ,2007. Article (CrossRef Link)

[16] Y. Zhang, D. Zhao and X. Ji, et al.. "A spatio-temporal auto-regressive model for frame rate up-conversion," *IEEE Transaction on Circuits System and Video Technology*, vol.19, no.9, pp.1289-1301, 2009. Article (CrossRef Link)

[17] Y. B. Zhang, D. B. Zhao and S. W. Ma, et al.. "A motion-aligned auto-regressive model for frame rate up-conversion," *IEEE Transaction on Image Processing*, vol.19, no.5, pp.1248-1258, 2010.

Article (CrossRef Link)

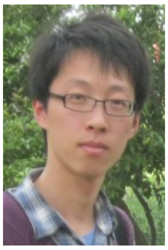
[18] H. Liu, R. Xiong and B. Zhao, et al.. "Multiple hypotheses Bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations," *IEEE Transaction on Circuits System and Video Technology*, vol.22, no.8, pp.1188-1198, 2012. Article (CrossRef Link)

[19] R. Li, Z. L. Gan, and Z. G. Cui, et al.. "Joint motion-compensated Interpolation using eight-neighbor block motion vectors." *IEICE Transaction on Information and System*, vol.E96-D, no.4, pp.976-979, 2013. Article (CrossRef Link)

[20] J. Ascenso, C. Brites and F. Pereira. "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *Proc. of 5th EURASIP Conference on Speech and Image Processing*, *Multimedia Communications and Services*, pp.1-6, July, 2005. Article (CrossRef Link)

[21] A. N. Tikhonov and V. Y. Arsenin, "*Solutions of Ill-Posed Problems*," V. H. Winston & Sons, Washington, D. C.. 1977. Article (CrossRef Link)

[22] J. Zhai, K. Yu and S Li. "A low complexity motion compensated frame interpolation method," in *Proc. of IEEE International Symposium on Circuits and Systems*, pp.4927-4930, 23-26 May, 2005. Article (CrossRef Link)



Ran Li is currently a Ph.D. candidate at Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include compressed sensing, distributed video coding and image communications.

Minghu Wu is currently a Ph.D. candidate at Nanjing University of Posts and Telecommunications, Nanjing, China. He is also an associate professor in the Hubei University of Technology. His research interests include compressed sensing and distributed video coding.

Zongliang Gan received his M.S. degree and the Ph.D. degree in signal and information processing from Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China, in 2004 and 2007 respectively. He is currently a lecturer in the Nanjing University of Posts and Telecommunications. His research interests include distributed video coding, image super-resolution reconstruction and image communications.

Ziguan Cui received his Ph.D. degree in signal and information processing from Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China, in 2011. He is currently a lecturer in the Nanjing University of Posts and Telecommunications. His research interests include video coding and image communications.

Xiuchang Zhu received his B.S. and M.S. degrees from Nanjing University of Posts and Communications in 1982 and 1987, respectively. He has been working in Nanjing University of Posts and Communications since 1987. At present, he is a Professor and the direct of Jiangsu Key Library of Image Processing and Image Communications. His current research interests focus on multimedia information, especially on the collection, processing, transmission and display of image and video.