A Spatio-Temporal Auto Regressive Model for Frame Rate Upconversion

Yongbing Zhang, Debin Zhao, Xiangyang Ji, Ronggang Wang, and Wen Gao, *Fellow, IEEE*

***Abstract*— This paper proposes a spatio-temporal auto regressive (STAR) model for frame rate upconversion. In the STAR model, each pixel in the interpolated frame is approximated as the weighted combination of a sample space including the pixels within its two temporal neighborhoods from the previous and following original frames as well as the available interpolated pixels within its spatial neighborhood in the current to-be-interpolated frame. To derive accurate STAR weights, an iterative self-feedback weight training algorithm is proposed. In each iteration, first the pixels of each training window in the interpolated frames are approximated by the sample space from the previous and following original frames and the to-be-interpolated frame. And then the actual pixels of each training window in the original frame are approximated by the sample space from the previous and following interpolated frames and the current original frame with the same weights. The weights of each training window are calculated by jointly minimizing the distortion between the interpolated frames in the current and previous iterations as well as the distortion between the original frame and its interpolated one. Extensive simulation results demonstrate that the proposed STAR model is able to yield the interpolated frames with high performance in terms of both subjective and objective qualities.**

***Index Terms*— Auto regressive model, frame rate upconversion, self-feedback, training window.**

I. Introduction

THE EXPLOSIVE growth of image sources and display devices in the consumer market, due to the rapid development of advanced television and multimedia techniques, has placed a high demand on the conversion between various video formats. Frame rate upconversion (FRUC) is a widely investigated technique to upconvert the temporal resolution of video sequences. FRUC has many applications, among which the most practical one is format conversion (for example from 24 frames/s film content to 30 frames/s video content, or from 50 fields to 60 fields/s). Besides format conversion, FRUC is also applicable to low-bitrate video coding [1], in which one can reduce the frame rate to the half or even lower ratio of its original, and then only encode the low rate frames with better visual quality. At the decoder side, FRUC can be performed to restore the original temporal resolution with higher visual quality. In addition, FRUC may be of great help for slow-motion playback and the rate allocation policy of a scalable video coding scheme [2].

Numerous FRUC algorithms have been developed to upconvert the frame rates. A straightforward FRUC is to simply combine adjacent video frames, e.g., frame repetition or frame averaging (FA) [1], in which object motion is not taken into account. Thus, this method works well only if there is little or no motion between adjacent frames. As the intensity of motion increases, frame repetition will cause jerk, and the resulting image will look very choppy or unsmooth. In addition, by applying FA, the image will look blurred when motion occurs, and sometimes ghost artifacts can be perceived.

Contrary to frame repetition or FA, another kind of FRUC methods performs frame interpolation along the motion trajectory to achieve better visual quality [1]–[19]. This method is called MC-FRUC (motion compensation-FRUC). Since the accuracy of motion estimation (ME) plays a significant role in MC-FRUC, many algorithms have been developed to derive more accurate motion vectors. Among them the block-matching algorithm (BMA) has broad application in MC-FRUC [3], [4]. Since motion vectors derived by BMA are often not accurate enough, several approaches for more faithful ME have also been proposed in recent works [5]–[7], [9]. Haan *et al*. [9] proposed a 3-D recursive search (3-DRS) method to obtain accurate motion vectors. Choi *et al*. [5] proposed a FRUC algorithm using bidirectional ME to derive more faithful motion vectors. A hierarchical MC technique was also proposed in [6] to achieve better visual quality. Both these methods in [5] and [6] outperform BMA. However, they are both based on the assumption of translational motion with constant velocity, which is not always true. In [7], 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 nonrigid objects.

Different from ME process in video coding [20], ME is performed with the absence of actual frames in FRUC.

Manuscript received January 31, 2008; revised June 14, 2008 and December 21, 2008. First version published May 12, 2009; current version published September 16, 2009. This work was partly supported by the National Science Foundation of China, 60736043 and 60833013, the Major State Basic Research Development Program of China, 973 Program 2009CB320905, and the France Telecom Project, OR_PED-BBR. This paper was recommended by Associate Editor J. Boyce.

Y. Zhang and D. Zhao are with the Department of Computer Science, Harbin Institute of Technology, Harbin 150001, China (e-mail: ybzhang@jdl.ac.cn; dbzhao@vilab.hit.edu.cn).

X. Ji is with the Broadband Networks and Digital Media Laboratory, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail:xyji@mail.tsinghua.edu.cn).

R. Wang is with France Telecom Research and Development, Beijing Company Ltd, Beijing 100190, China (e-mail: ronggang.wang@orange-ftgroup.com).

W. Gao is with the Key Laboratory of Machine Perception, the School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: wgao@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2022798

Hence, the derived motion trajectory may be not consistent sometimes. To resolve such a problem, many motion vector postprocessing methods are proposed to smooth the motion field and improve the subjective perception [3], [7], [8]. What is more, for the areas with small objects, irregular shaped objects, and object boundaries, fixed size block MC usually does not work well. To deal with this situation, variable size block MC was proposed to reconstruct the edge information with higher quality [13]. Furthermore, overlapped block MC (OBMC) [21] is applied to suppress the blocking artifacts that are usually observed when a block has a significantly different motion vector compared with its neighboring blocks. However, OBMC may sometimes oversmooth the edges of the image and thus degrade the image quality. To reduce the oversmooth effect of OBMC, Choi *et al*. [13] 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.

So far, all these algorithms are still unsatisfactory for FRUC practical applications due to the aforementioned underlying problems. Auto regressive (AR) model [22], which is able to estimate the noise covariance function even on very short record of data, may give us some inspiration to further improve the quality of the interpolated frames. AR has been applied in many image processing applications, such as detecting and interpolating "dirt" areas in image sequences [23], [24], ME [25], super-resolution [26], forecasting of spatio-temporal data [27], as well as backward adaptive video coding [28]. All these AR models achieved good performance in their corresponding fields, due to the superior properties of exploiting redundancies in the spatial or temporal domains. Since the quality of the interpolated frames in FRUC heavily relies on the extent of redundancy exploiting, it would be very desirable to apply AR in FRUC.

Based on the inspirations mentioned above, in this paper a spatio-temporal AR (STAR) model is proposed to perform FRUC. By extending our previous work [29], in the STAR model each pixel in the interpolated frame is approximated as the weighted combination of a sample space including the pixels within its two temporal neighborhoods from the previous and following original frames as well as the available interpolated pixels within its spatial neighborhood in the current to-be-interpolated frame. In addition, an iterative self-feedback weight training method is proposed to derive accurate STAR weights. In each iteration, first the pixels of each training window in the interpolated frames are approximated as a weighted combination of the sample space including the pixels within its two temporal neighborhoods from the previous and following original frames as well as the available interpolated pixels within its spatial neighborhood in the current to-be-interpolated frame. Then the actual pixels of each training window in the original frame are approximated by the sample space from the previous and following interpolated frames and the current original frame with the same weights. The STAR weights for each training window are finally derived by jointly minimizing the distortion between the interpolated frames in the current and previous iterations as well as the distortion between the original frame and its approximated frame. Due to the property of the self-feedback weight training, more accurate weights can be obtained. Consequently, the STAR model is able to consider the nonstationary statistics of video signals, and thus can resolve the challenging problems, such as zooming, panning and nonrigid objects, quite well. In addition, the temporal qualities of the interpolated frames are improved, since the interpolated pixels are generated as a weighted summation of the pixels within a spatio-temporal neighborhood with more accurate weights and thus can make the interpolated frames much smoother.

The remainder of this paper first gives a brief introduction of the related works in Section II, including traditional MC-FRUC methods as well as AR models. In Section III, the detailed description of the proposed STAR model is presented. The self-feedback weight training method is presented in Section IV. The experimental results and analysis are provided in Section V. Finally, this paper is concluded in the last section.

II. Related Works

In this section we will give a brief introduction on the related works: MC-FRUC and auto regressive model.

A. MC-FRUC

MC-FRUC outperforms the frame repetition and FA methods due to the exploiting of motions between successive frames. In MC-FRUC, each frame to be interpolated is divided into blocks, and then each block is motion-compensated using the motion information related to the previous and following frames. According to the type of motion compensation, MC-FRUC can be categorized into motion compensation interpolation (MCI), OBMC [14], and AOBMC [13].

Let $\hat{F}_t(x, y)$, $F_{t-1}(x, y)$, and $F_{t+1}(x, y)$ denote the pixels in the interpolated frame, the previous, and the following frames located at spatial location (x, y), respectively. Then, the interpolation by MCI can be expressed as

$$\hat{F}_t(x, y) = \frac{1}{2}\left(F_{t-1}\left(x + v_x, y + v_y\right) + F_{t+1}\left(x - v_x, y - v_y\right)\right) \quad (1)$$

where v_x and v_y represent the motion vectors pointing to the previous frame in the horizontal and vertical directions, respectively. From (1), we can see that in MCI, each block is assumed to experience a translational motion. However, since some objects in video sequences may be irregular and the adjacent blocks may have significantly different motion vectors, serious blocking artifacts may be perceived in the interpolated frames generated by MCI.

To reduce the blocking artifacts, OBMC [14] was proposed by positioning overlapped blocks from a reference frame using a weighting window. As illustrated in Fig. 1, suppose the top left four neighboring blocks V1, V2, V3, and V4 have separate motion vectors (v_{1x}, v_{1y}), (v_{2x}, v_{2y}), (v_{3x}, v_{3y}), and (v_{4x}, v_{4y}), respectively. Then, the pixels in region A, which

overlaps the four blocks V1, V2, V3, and V4, are interpolated by

$$\hat{F}_t(x,y) = \left(\sum_{i=1}^{4} F_{t-1}\left(x+v_{ix}, y+v_{iy}\right) + F_{t+1}\left(x-v_{ix}, y-v_{iy}\right)\right) \Big/ 8. \quad (2)$$

The pixels in region B, which overlaps two blocks V3 and V4, are interpolated by

$$\hat{F}_t(x,y) = \left(\sum_{i=3}^{4} F_{t-1}\left(x+v_{ix}, y+v_{iy}\right) + F_{t+1}\left(x-v_{ix}, y-v_{iy}\right)\right) \Big/ 4. \quad (3)$$

The pixels in region C, which only overlaps one block V4, are interpolated by

$$\hat{F}_t(x,y) = \left(F_{t-1}\left(x+v_{4x}, y+v_{4y}\right) + F_{t+1}\left(x-v_{4x}, y-v_{4y}\right)\right)/2. \quad (4)$$

When motion activities are low, OBMC can effectively reduce blocking artifacts and provide good visual quality. However, OBMC may yield blurring or oversmoothing artifacts if the adjacent blocks have substantially different motions.

To overcome the shortcomings of OBMC, Choi *et al.* [13] proposed an AOBMC, in which the weighting coefficients are adjusted adaptively according to the reliabilities of the neighboring motion vectors. In their work, the reliability of the neighboring motion vector $v_{i+p,j+q}$ for the prediction of the current block $B_{i,j}$ is defined as

$$\Phi_{B_{i,j}[v_{i+p,j+q}]} = \frac{SBAD\left[B_{i,j}, v_{i,j}\right]}{SBAD\left[B_{i,j}, v_{i+p,j+q}\right]} \quad (5)$$

where $SBAD\left[B_{i,j}, v_{i,j}\right]$ represents the sum of bilateral absolute difference when applying $v_{i,j}$ for current block $B_{i,j}$. It achieves better visual quality than that generated by OBMC for the majority of cases. However, for some complex motions, such as zooming, rotation, and local deformation, AOBMC is still not able to represent them well.

B. MC-FRUC Auto-Regressive Model

AR model has been widely investigated in image and video applications. In [23], [24], AR was exploited to detect and interpolate the missing data in images based on its property of handling the fine detail and accounting for intensity variation. In [25], [26], AR was exploited to perform ME and super-resolution, where each pixel was predicted to be a linear combination of pixels in a spatial neighborhood in the same frame. AR was also used in data acquisition: e.g., an AR model concerning spatio-temporal data with a short observation history was proposed for forecasting data [27], in which each pixel was predicted as the linear combination of neighboring pixels in the previous frame. In [28], Li proposed a backward adaptive video coding exploiting AR model, where each pixel was predicted not only from the pixels in the previous frames but also from the pixels in the current frame.

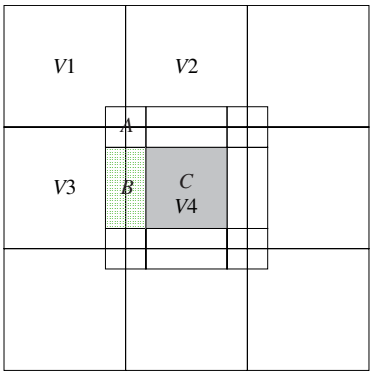


Fig. 1. Illustration of OBMC.

All these an AR models have made a great success in the above-mentioned fields. And it seems to be a good choice to apply an AR model to enhance the quality of the interpolated frame in FRUC. However, if we apply an AR model in FRUC, two problems must be solved: how to approximate the actual pixels in the interpolated frames by more efficiently exploiting the information in the spatio-temporal domains, and how to derive more accurate weights with the absence of the actual pixels in the interpolated frame. For the first problem, we propose that each pixel in the interpolated frame is approximated as a weighted combination of a sample space including the pixels within its two temporal neighborhoods from the previous and following original frames as well as the available interpolated pixels within its spatial neighborhood in the current to-be-interpolated frame. For the second problem, a self-feedback weight training method is proposed to derive more accurate weights.

III. Spatio-Temporal Auto-Regressive Model

In the proposed STAR model, each to-be-interpolated frame is divided into nonoverlapped training windows, each of which is trained separately and different weights are applied to different training windows. In each training window with the size $W_x \times W_y$, the pixel located at (k,l) will be interpolated by the STAR model as

$$\begin{aligned}\hat{R}_{t-1}(k,l) = &\sum_{-L\le(u,v)\le L}\sum R_{t-2}(k+u, l+v)\times W_p(u,v) \\ &+ \sum_{-L\le(u,v)\le L}\sum R_t(k+u, l+v)\times W_f(u,v) \\ &+ \sum_{\substack{\{v<0,-L\le u\le L\}\\ \cup\{v=0,u<0\}}}\sum \hat{R}_{t-1}(k+u, l+v)\times W_s(u,v)\end{aligned} \quad (6)$$

where $\hat{R}_{t-1}(k,l)$ represents the interpolated pixel at (k,l) in the training window $\hat{R}_{t-1}$, $R_{t-2}(k,l)$ and $R_t(k,l)$ represent the actual pixels in the original low rate frames at time instances $t-2$ and t, respectively, and L is the spatio-temporal order of the STAR model. Here $W_p(u,v)$ and $W_f(u,v)$ represent the STAR weights corresponding to the temporal neighborhoods, denoted by the solid circle in Fig. 2, in the previous and following frames, respectively. $W_s(u,v)$ represents the

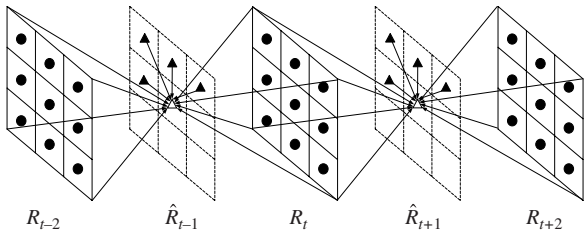


Fig. 2. STAR model with spatio-temporal order $L = 1$. The solid circle and triangle represent the actual pixel and interpolated pixel, respectively.

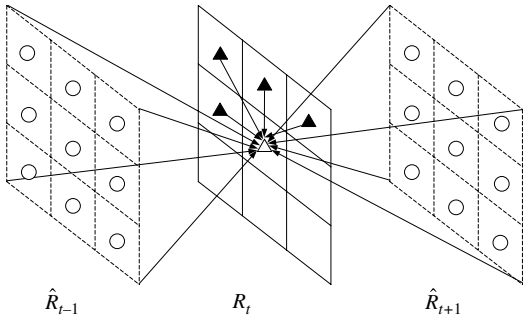


Fig. 3. Self-feedback interpolation for the weight training. The hollow circle represents the already interpolated pixel within the to-be-interpolated frames $t-1$ and $t+1$, and the solid triangle represents the approximated pixel within the original frame t.

STAR weight corresponding to spatial neighborhood, denoted by the solid triangle in Fig. 2, in the current to-be-interpolated frame. Note that due to the piecewise stationary characteristics of natural image, the STAR weights are assumed to be the same for all the pixels within each training window $\hat{R}_{t-1}$.

According to (6), the quality of the to-be-interpolated frames heavily relies on the accuracy of the STAR weights. The optimum STAR weights can be easily computed by minimizing the mean square error between the actual and the interpolated pixels according to

$$\begin{aligned}\varepsilon &= E\left(R_{t-1} - \hat{R}_{t-1}\right) \\ &= \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} E\left[\left(R_{t-1}(k,l) - \hat{R}_{t-1}(k,l)\right)^2\right] \end{aligned} \tag{7}$$

where $\hat{R}_{t-1}(k,l)$ represents the interpolated pixel within the training window $\hat{R}_{t-1}$ and $R_{t-1}(k,l)$ represents the corresponding actual pixel in the original frame $t-1$. However, due to the absence of the corresponding actual pixel in the original frame $t-1$ in FRUC, (7) cannot be used to derive these optimum STAR weights. To address this issue, a self-feedback weight training algorithm is proposed in the following section.

IV. Self-Feedback Weight Training

The self-feedback weight training consists of two stages. In the first stage, the pixels in the training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$, as the solid triangles shown in Fig. 2, are interpolated according to (6). Then in the second stage, the actual pixels in the corresponding window in frame t are also approximated by the pixels in the two already interpolated training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$ as well as the available interpolated pixels in the current training window $\hat{R}_t$ with the same weights as in the first stage (see Fig. 3). This approximation can be described as follows:

$$\begin{aligned}\hat{R}_t(k,l) = &\sum_{-L\le(u,v)\le L}\sum \hat{R}_{t-1}(k+u, l+v)\times W_p(u,v) \\ &+ \sum_{-L\le(u,v)\le L}\sum \hat{R}_{t+1}(k+u, l+v)\times W_f(u,v) \\ &+ \sum_{\substack{\{v<0,-L\le u\le L\}\\ \cup\{v=0,u<0\}}}\sum \hat{R}_t(k+u, l+v)\times W_s(u,v).\end{aligned} \tag{8}$$

One direct way to derive the optimal weight is to minimize the distortion between the actual pixels and the corresponding approximated pixels within the training window $\hat{R}_t$. However, it will result in groups of multiple quadratic equations, which is a nonlinear optimization problem (see Appendix A for detailed discussion).

To get rid of the problem of multiple quadratic equations, we instead devise an iterative method using a linear least square method. If we rewrite the STAR weights in a 1-D manner, then the weight vector after the ith iteration can be defined as $\vec{W}^i = \left[W_p^i, W_f^i, W_s^i\right]^T$. Here W_p^i represents the weight vector of the previous temporal neighborhood, W_f^i represents the weight vector of the following temporal neighborhood, and W_s^i represents the weight vector of the spatial neighborhood. Assume we have obtained the interpolated pixels $\hat{R}_{t-1}^i(k,l)$ and $\hat{R}_{t+1}^i(k,l)$ within the training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$ prior to the ith iteration. $\vec{W}^i$ can be computed by the closed-form of the least-square algorithm as

$$\vec{W}^i = \left(A^i\right)^{-1} \times \left(B^i\right) \tag{9}$$

where A^i is a matrix, and B^i is a column vector (see Appendix B for details on computing A^i and B^i).

After we obtain $\vec{W}^i$, the newly interpolated pixels $\hat{R}_{t-1}^{i+1}(k,l)$ and $\hat{R}_{t+1}^{i+1}(k,l)$ are computed according to (6), and $\hat{R}_t^{i+1}(k,l)$ is computed according to (8). To make a decision whether the FRUC will be terminated in the current iteration, we compute the distortion of the interpolated pixels within the training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$ between two successive iterations as well as the distortion between the actual and

TABLE I
Summary of The Proposed Self-Feedback Weight Training Algorithm

initialize $\hat{R}_{t-1}^0$ and $\hat{R}_{t+1}^0$
for $i = 0$ to iMax-1
  Compute A^i and B^i according to (18a) and (19a);
  Compute the weight vector $\vec{W}^i$ according to (9);
  Compute $\hat{R}_{t-1}^{i+1}$ and $\hat{R}_{t+1}^{i+1}$ according to (6) by $\vec{W}^i$;
  Compute $\hat{R}_t^{i+1}$ according to (8) by $\vec{W}^i$;
  Compute $D(i)$ according to (10);
  if $D(i) < Threshold$
    break;
  end if
end for

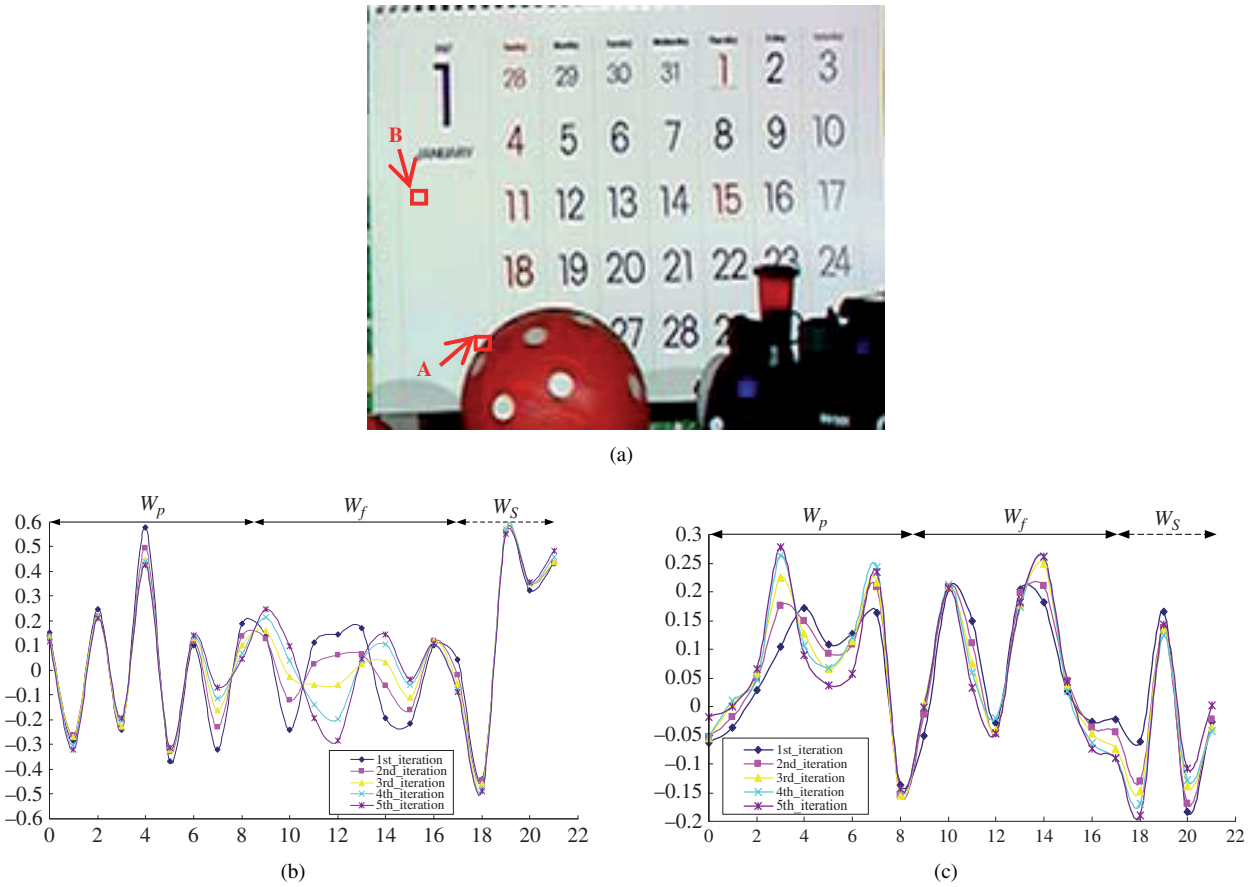


Fig. 4. Illustration of spatio-temporal adaptation over iterations for the second interpolated frame of *Mobile* (CIF). (a) Second interpolated frame of *Mobile* indicated by training windows A and B with and without occlusions. (b) and (c) STAR weight profiles for training windows A and B. In (b) and (c), the vertical axis represents the weight value, and the horizontal axis represents the serial number of the weights in the two temporal neighborhoods and the spatial neighborhood, which are arranged in the concatenated and lexicographically order. The two solid regions denote the previous and following temporal neighborhoods, and the dashed region denotes the spatial neighborhood.

interpolated pixels within the training window $\hat{R}_t$. In other words, the total distortion is jointly obtained as follows:

$$\begin{aligned} D(i) = &\sum_{k=0}^{W_x}\sum_{l=0}^{W_y} E\left[\left(\hat{R}_{t-1}^{i+1}(k,l) - \hat{R}_{t-1}^{i}(k,l)\right)^2\right] \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} E\left[\left(\hat{R}_{t+1}^{i+1}(k,l) - \hat{R}_{t+1}^{i}(k,l)\right)^2\right] \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} E\left[\left(\hat{R}_{t}^{i+1}(k,l) - R_t(k,l)\right)^2\right] \end{aligned} \quad (10)$$

where $\hat{R}_{t-1}^{i}$ and $\hat{R}_{t-1}^{i+1}$ represent the interpolated training windows prior to and after the ith iteration, respectively. If $D(i)$ is smaller than a preset threshold, or i is larger than the predefined maximum iteration number, the iteration is terminated and $\vec{W}^i$ is set to be the ultimate weight vector of the STAR model. Thus, the interpolated pixels generated by $\vec{W}^i$ are considered to be the final interpolated pixels within training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$. Otherwise, i is increased by 1 and the self-feedback weight training algorithm is moved to the next iteration.

The summary of the proposed self-feedback weight training algorithm is illustrated in Table I. The initial values $\hat{R}_{t-1}^{0}$ and $\hat{R}_{t+1}^{0}$ can be computed by several methods, such as FA, MCI, OBMC, AOBMC etc., and the influences of different initial values will be analyzed in Section V.

Most of the computational complexity is concentrated on the calculation of matrix A^i when computing $\vec{W}^i$. In the proposed self-feedback weight training method, $O\left(L^4 * W_x * W_y\right)$ arithmetic operations are required to compute A^i if implemented straightforwardly. Such prohibitive computational cost is the major disadvantage of the self-feedback weight training method. However, the symmetric property of matrix A^i can be exploited to reduce the computational complexity. Furthermore, there exist fast algorithms for calculating A^i by exploiting the overlap of the blocks between adjacent pixels. For instance, the fast algorithm proposed by Wu [22] can greatly reduce the computational complexity of matrix A^i. With fast algorithms and more powerful computing resources, the running time of the self-feedback weight training can be further reduced.

V. Experimental Results and Analysis

Various standard video sequences with different sizes (QCIF, CIF, 4CIF, and 720P) have been tested in an attempt to shed some light on the quality of interpolated frames. To compare the performance, every other frame from test

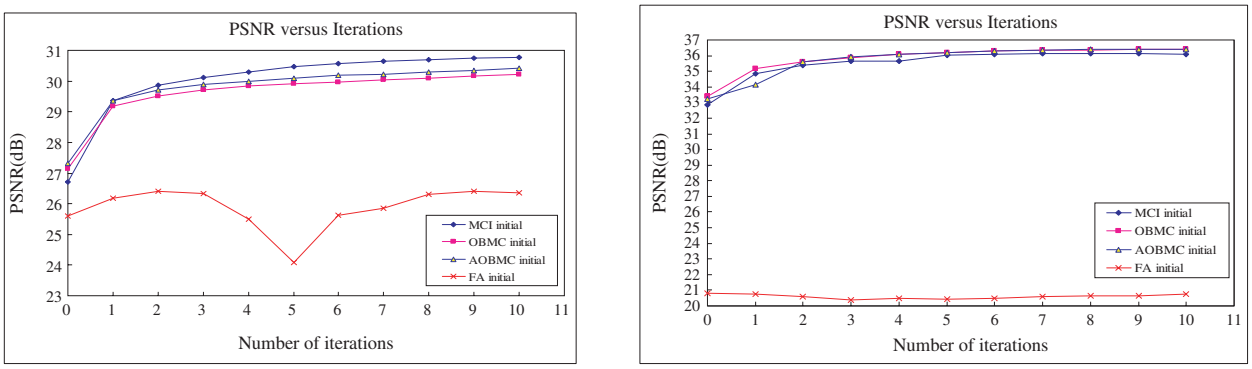


Fig. 5. PSNRs of the interpolated frames versus iteration numbers under different initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$. (a) Second interpolated frame of *Mobile* (CIF) and (b) first interpolated frame of *Flower* (CIF).

TABLE II
PSNRS OF INTERPOLATED FRAMES BY DIFFERENT METHODS

Resolution	Sequences	Input frames/s	Output frames/s	3-DRS	MCI (4 × 4)	MCI (8 × 8)	OBMC	AOBMC	STAR
QCIF	*Mobile*	15	30	33.170	33.472	33.360	33.791	34.251	**36.192**
	Foreman	15	30	34.051	37.721	38.600	38.756	38.778	**39.666**
CIF	*City*	15	30	25.987	32.603	33.940	34.353	34.562	**34.833**
	Flower	15	30	19.779	30.571	31.530	31.986	31.890	**33.354**
	Mobile	15	30	25.310	26.778	27.460	28.164	28.738	**29.487**
	Tempete	15	30	30.046	30.210	30.400	30.603	30.741	**30.861**
	Bus	15	30	18.778	25.801	26.132	26.684	26.999	**27.269**
4CIF	*City*	30	60	26.269	28.516	28.226	28.864	28.923	**30.132**
	Flower	12.5	25	18.495	28.292	28.160	28.473	28.521	**28.704**
	Mobile	12.5	25	20.724	25.475	25.822	26.220	26.365	**26.747**
720p	*Spincalendar*	30	60	25.858	26.971	27.630	28.241	28.281	**29.512**
	Sheriff	30	60	37.352	37.798	37.946	38.149	38.166	**38.081**
	City	30	60	30.554	30.019	30.267	30.563	30.639	**31.664**

sequences is skipped and interpolated by the STAR model as well as by 3-DRS, traditional MCI method, OBMC, and AOBMC. The interpolated frames are then compared with the accurate ones skipped in original sequences.

A. Weight Analysis

In this section we use a typical test sequence *Mobile* (CIF) to illustrate the spatio-temporal adaptation behaviors of the algorithm for regions with and without occlusions. The spatio-temporal order L is set to be 1 and the MCI results are used to generate the initial interpolations in this experiment. Two training windows are highlighted in Fig. 4(a). Training window A is the occluded area where temporal prediction does not work, and B is located in non-occluded areas. We can find in Fig. 4(b) that the self-feedback weight training method assigns larger weights to the previous temporal neighborhood and the spatial neighborhood and smaller weight values to the following temporal neighborhood. This is because training window A can be totally observed in the previous frame while only a part of it can be observed in the following frame due to the movement of the rolling ball. In contrast, we can find in Fig. 4(c) that larger weights are assigned to the previous and following temporal neighborhoods, and smaller weights are assigned to spatial neighborhood. This is because training window B can be entirely observed in both the previous and following frames. Such observations illustrate the adaptation of the self-feedback weight training method to spatial and temporal coherences.

Another observation is that the fluctuation of the majority weight values both in Fig. 4(b) and (c) moves toward the same direction across iterations. For example, in Fig. 4(b) the weights located at 4, 8, 11, 12, and 17 become smaller and smaller, while the weights located at 7, 10, 14, and 15 become larger and larger with iteration. Similarly, in Fig. 4(c) the weights located at 4, 5, 11, 16, 17, and 18 become smaller and smaller, while the weights located at 2, 3, 14, and 20 become larger and larger with iteration. The observations in Fig. 4(b) and (c) indicate that the fluctuation trend of the majority weights is the same, which can verify, to some extent, that the weights tend to converge with iteration.

B. Convergence Study

In this section, several experiments are conducted to study the effects of the initial pixel values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$ on the convergence of the proposed self-feedback weight training method. The convergence of the proposed self-feedback weight training method was evaluated using different initial

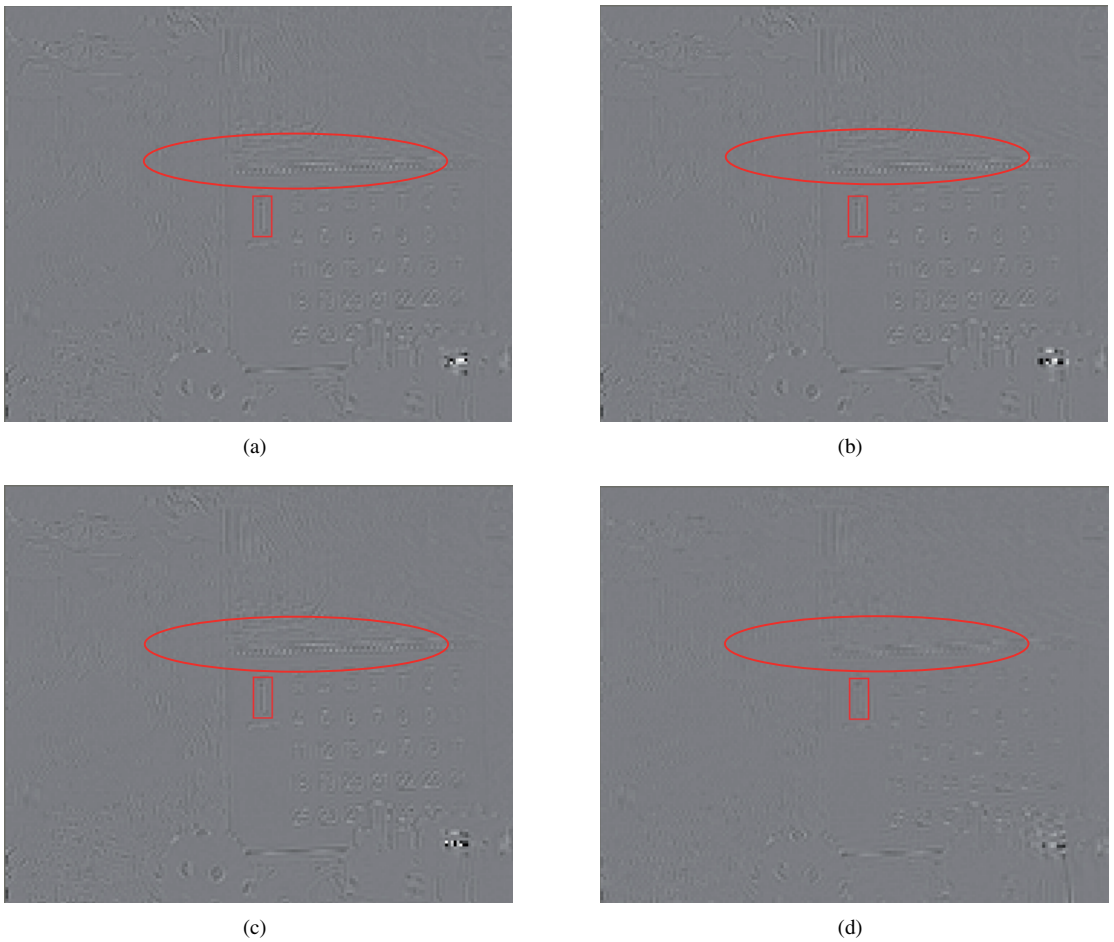


Fig. 6. Residual frame of the 16th frame within *Mobile* (QCIF) interpolated by different methods: (a) MCI, (b) OBMC, (c) AOBMC, and (d) proposed STAR model.

values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$ under the same training windows and the same spatio-temporal orders.

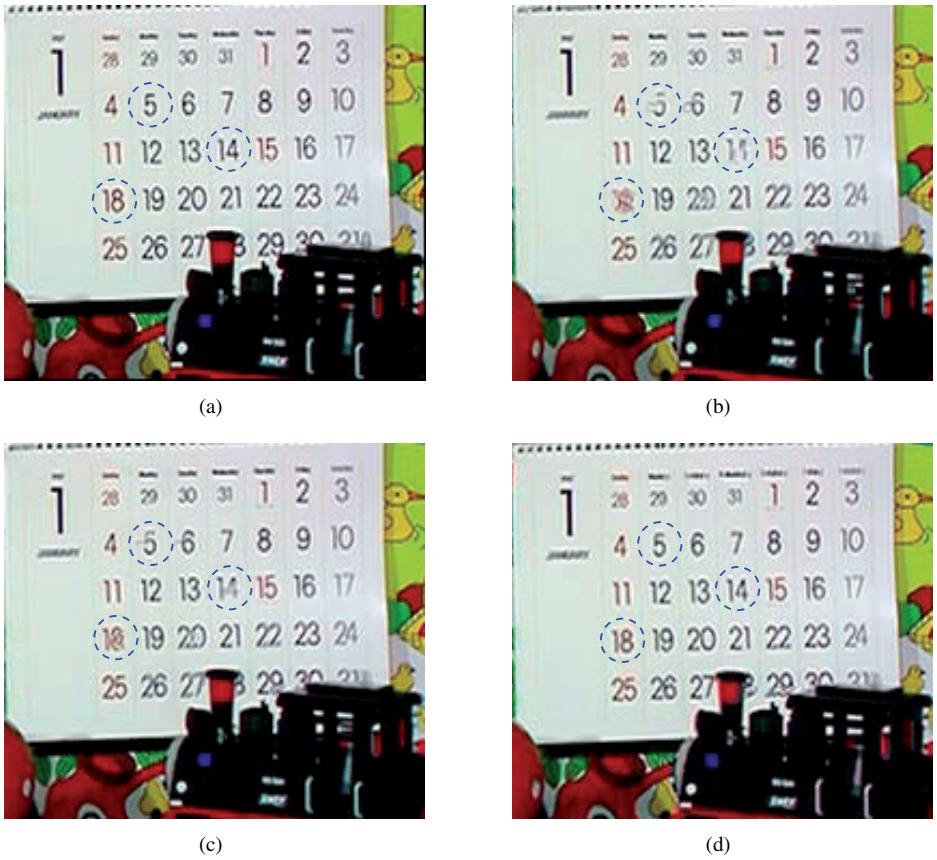
Four methods: FA, MCI, OBMC [14], and AOBMC [13], are performed to generate the initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$. In MCI, the bilateral ME as described in [14] is first performed, and then the motion vector postprocessing is applied to smooth the motion field, and finally the intermediate frame is interpolated by the smoothed motion vectors. Here the integer-pixel full search is first conducted, and then the half-pixel and the quarter-pixel accuracy search are processed and thus the motion vector of MCI is of quarter-pixel accuracy.

The PSNRs of the interpolated frames, by the proposed STAR model with different initial values, against the number of iterations are plotted in Fig. 5. It is observed that except for FA, the experiments with $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$ generated by MCI, OBMC, and AOBMC, all tend to converge with iteration. The PSNRs of the interpolated frames by FA are not improved with the iteration. This is because the initial values generated by FA are unreliable due to the neglect of motions between successive frames. This also reflects that the proposed weight training method will achieve bad performance if the initial values are too unreliable. However, if proper initial values are given, e.g., generated by MCI, OBMC, or AOBMC, the performance will be improved with iteration and the improvement becomes trivial after several iterations. The elegant performance is largely attributed to the self-feedback properties of the proposed weight training method, where the accuracy of the weights can be improved according to the feedback from the previous iteration, and thus is able to approach the optimum weights with the iteration. This provides empirical evidence on the convergence of the proposed self-feedback weight training method and an indication on the rationality of the initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$.

A check mechanism can be applied to see whether the proposed STAR model converges or not. In each iteration, the distortion $D(i)$ is compared to that in the previous iteration. If $D(i)$ becomes larger, we simply take the STAR model as not converging. In this case, the MCI results are taken as the final interpolated results. Note that if (9) has no solution, in case when $A(i)$ is a singular matrix, the MCI results are also used as the final interpolated results. Obviously, results generated by other interpolation methods, e.g., OBMC and AOBMC, can also be used to replace the STAR model when it fails.

C. Parameter Selection

As indicated in the previous section, if we choose proper initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$, e.g., generated by MCI, OBMC, and AOBMC, the proposed weight training method tends to converge when the iteration number is big enough, and thus



(a) (b) (c) (d)

Fig. 7. Visual comparisons of part of the original frame and interpolated frames of the 18th frame within *Mobile* (CIF). (a) Original frame, (b) OBMC, (c) AOBMC, and (d) proposed STAR model.

in the following experiments the MCI results are set to be the initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$ before the iteration of the self-feedback weight training begins. Four parameters of the STAR model: the size of training window, the spatio-temporal order, the maximum iteration number, and the total distortion preset threshold, should be set properly. If the size of training window is too small, there are not enough pixel samples to derive more accurate weights. On the other hand, if the size of training window is too large, it does not take into account the variation properties of image signals. By our observation, the size of training window is set to be 16×16 for QCIF sequences, and for the higher resolution (CIF, 4CIF, 720P) sequences it is set to be 32×32 empirically in our experiments. The total distortion preset threshold is set to be 50.

Another observation is that the spatio-temporal order is closely related to the motion in the training window. That is because a smaller spatio-temporal order will achieve good performance for stationary regions; however, for the moving regions larger spatio-temporal order is necessary. Since the MCI results are set to be the initial values $\hat{R}^0_{t-1}$ and $\hat{R}^0_{t+1}$, the motion vector in MCI can be utilized to measure the motion in the training window. In our experiment, the spatio-temporal order of the STAR model is computed by

$$L = \max_{\text{block}_j \in R} \left\{ abs\left(\left\lfloor mvx_{\text{block}_j} \right\rfloor\right), abs\left(\left\lfloor mvy_{\text{block}_j} \right\rfloor\right) \right\} + 1 \tag{11}$$

where R represents the training window, mvx_{block_j} and mvy_{block_j} represent the horizontal and vertical motion vectors of the jth block when performing MCI, and $\lfloor \rfloor$ is the floor operator, which maps mvx_{block_j} or mvy_{block_j} with sub-pixel accuracy to the next full-pixel position. It should be noted that for sequences with fast motion, the spatio-temporal order will be quite large according to (11), and thus it may introduce outliers into the optimization process, which may prevent the training algorithm from finding the correct weighting vector. To avoid this, the maximum L is set as 6 in the experiment.

We also found that when the maximum iteration number exceeds 4, the improvement is usually very trivial. Therefore, in our experiment the maximum iteration number is set as 4 empirically.

D. Objective and Subjective Evaluation

To demonstrate the performance of the proposed STAR method, we compare it with 3-DRS method [9], traditional MCI method, OBMC [14], and AOBMC [13].

In 3-DRS, motion vectors are estimated by the 3-D recursive block matching [9], and the holes and overlapping regions are processed using the method in [10] and [11]. To show the impact of the block size in MCI, we implemented the MCI method with the block size 4×4 and 8×8, respectively (shown in Table II). We found that, different from the video coding [20], smaller block size, e.g., 4×4, does not ensure better performance. Theoretically, small block sizes should yield better performance. However this requires very accurate ME, called "true ME." But finding a "true ME" is a major

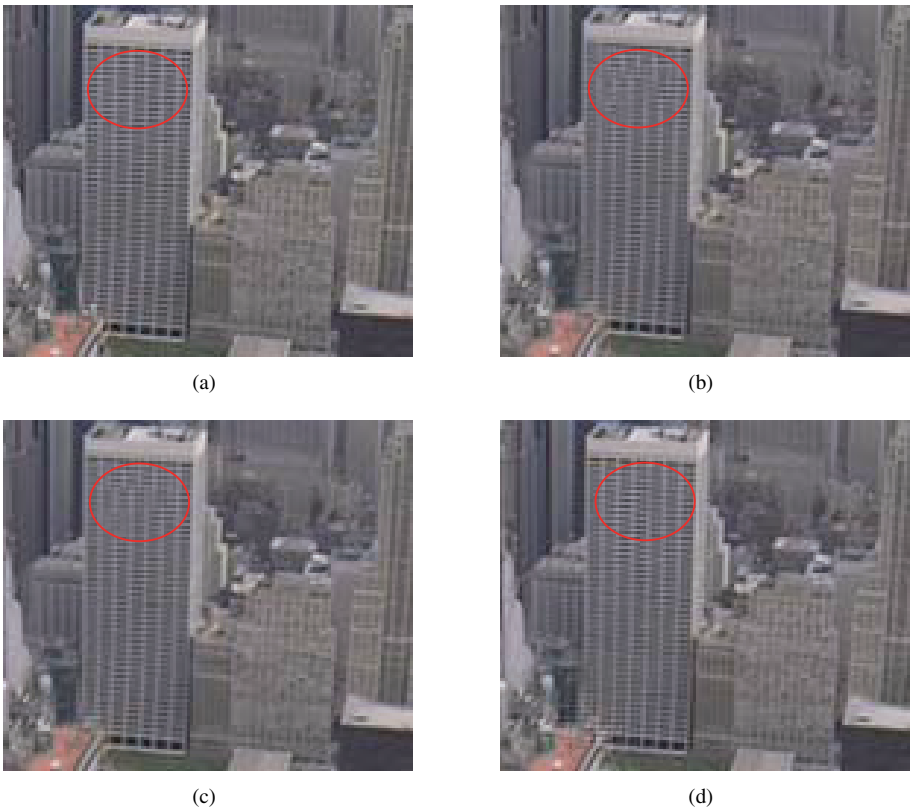


Fig. 8. Visual comparisons of part of the original frame and the interpolated frames of the 14th frame within *City* (CIF). (a) Original frame, (b) OBMC, (c) AOBMC, and (d) proposed STAR model.

problem in itself, especially in FRUC, where the actual pixels in the to-be-interpolated frame are not available. Based on such observations, the block size is set to 8×8 for MCI, and correspondingly, the block size for OBMC and AOBMC is also set to 8×8. For each test sequence, 50 frames are skipped and interpolated by the STAR model and all the compared methods. The resolution, input frames/s, output frames/s, and average PSNRs of the interpolated frames generated by the STAR model as well as other methods are depicted in Table II. It can be seen that except for *City* (720p), 3-DRS method has worse performance than other methods, partly due to the feet that there are fewer positions searched for 3-DRS compared with other methods. For all the test sequences, OBMC outperforms MCI due to the application of multihypothesis and thus can alleviate the artifacts caused by MCI. Except for *Flower* (CIF), AOBMC achieves higher PSNR than OBMC, since it is capable of adjusting the weights adaptively according to the reliability of neighboring blocks. However, the improvement of AOBMC against OBMC is insignificant for some sequences, i.e., *Foreman* (QCIF) and *Spincalendar* (720p), for which the PSNR gain is less than 0.1 dB. On the contrary, except for *Sheriff* (720p), the performance improvement of the STAR model is significant when compared with other methods. Especially for *Mobile* (QCIF) and *Flower* (CIF), the PSNR gains, compared with MCI, are up to 2.832 and 1.824 dB, respectively. Besides, for *City* (4CIF) and *Spincalendar* (720p), the PSNR gains are up to 1.9 dB. This is because *Flower* (CIF) is full of nonrigid objects and consequently could not be approximated well based on the assumption that all the pixels within one block have unique motion, which is the foundation of traditional FRUC methods. And for the other three test sequences, there are a lot of camera motions, i.e., rotation in *Mobile* (QCIF), panning in *City* (4CIF), and spinning in *Spincalendar* (720p), which are very hard to be represented by traditional FRUC methods, e.g., 3-DRS, MCI, OBMC, and AOBMC.

Fig. 6 depicts the residual frames of the 16th interpolated frames within *Mobile* (QCIF) yielded by MCI, OBMC, AOBMC, and the proposed STAR model. It is easy to observe that the residual energies yielded by both MCI and OBMC are large although OBCM is able to yield less residual energy. AOBMC has less residual energy compared to the former two methods; nevertheless, the improvement is still not significant enough. In contrast, the proposed STAR model provides the least residual energy for most regions. Especially for the regions marked by the red rectangle the contour of number "1" can be easily perceived in the residual frame yielded by MCI, OBMC, and AOBMC methods. However, it is hard to be perceived in the residual frame yielded by the STAR model. Besides, for the region marked by the ellipse, there are significant energies in the residual frames yielded by MCI, OBMC, and AOBMC methods. Nevertheless, the residual energies in the region marked by the ellipse are greatly reduced by the STAR model.

Fig. 7 illustrates part of the original frame and the interpolated frames of the 18th frame within *Mobile* (CIF) by OBMC, AOBMC, and the proposed STAR model. The interpolated frame by OBMC exhibits the worst visual quality, where some numbers in the calendar cannot be observed clearly, e.g.,

numbers "5," "14," and "18," as marked by the dashed circles in Fig. 7(b). It is mainly due to the fact that the contours of the numbers are not aligned with the block, and even OBMC cannot alleviate the ghost artifacts. The interpolated frame by AOBMC has better visual quality than the one interpolated by OBMC, e.g., numbers "5," and "14," as marked by the dashed circles in Fig. 7(c), can be easily perceived. This is mainly attributed to the ability of adaptively adjusting the weights in AOBMC. Nevertheless, number "18," marked by the dashed circle in Fig. 7(c), still cannot be perceived. This is because the neighboring motion vectors around number "18" are not reliable and consequently the ability of adjusting the weights is impaired. On the other hand, the proposed STAR model alleviates the ghost artifacts more efficiently and provides the best image quality, e.g., besides numbers "5" and "14," number "18," as marked by the dashed circles in Fig. 7(d), can also be easily perceived. This mainly benefits from the proposed STAR model's ability of tuning the STAR weights according to the characteristics of its spatio-temporal neighborhoods.

Fig. 8 exhibits the visual performance of part of the 14th interpolated frame within *City* (CIF). The proposed STAR model achieves the best visual quality compared with OBMC and AOBMC methods. Especially, for the regions marked by the red ellipse, prominent blocking artifacts can be perceived in the frames yielded by OBMC and AOBMC, whereas no obvious blocking artifacts are observed in the part of frame yielded by the STAR model. That is because the motion vectors in the marked region are too irregular, and even OBMC cannot effectively suppress the blocking artifacts. AOBMC can alleviate the blocking artifacts to some extent; nevertheless, it could not remove the blocking artifacts thoroughly since the motion vector reliabilities of neighboring blocks are not high enough. In contrast, due to the ability of adaptively tuning STAR weights according to the local spatio-temporal characteristics, the interpolated frame yielded by the STAR model exhibits the best visual quality.

VI. Conclusion

In this paper, a spatio-temporal auto regressive (STAR) model has been proposed for frame rate upconversion. In the STAR model, the spatio-temporal interactions among pixels were exploited, where each pixel is interpolated as the weighted summation of the pixels within its spatio-temporal neighborhoods. To derive more accurate STAR weights with the absence of actual pixels in the to-be-interpolated frame, a self-feedback weight training method was proposed. Applying the self-feedback weight training method, the proposed STAR model is able to take advantage of the nonstationary statistics of video signals, and thus can resolve the challenging problems such as zooming, panning, and nonrigid objects quite well.

Extensive experiments have been performed on various test sequences with different resolutions to demonstrate the validity of the proposed STAR model. Both subjective and objective evaluations of the proposed STAR model were carried out. Experimental results have demonstrated that the proposed STAR model has superior performance to traditional FRUC methods in terms of both objective and subjective criterions. The shortcoming of the STAR model is the high computation complexity compared with other FRUC methods. Nevertheless, the complexity can be reduced by fast algorithms. Besides, it is applicable for offline FRUC applications to achieve high-quality interpolated frames.

Appendix A

For simplicity of discussion, the spatial weight vector of the STAR model is assumed to be zero. The optimal weight vector derivation process by directly minimizing the actual and the interpolated pixels within training window $\hat{R}_t$ can be described as follows. First, the interpolated pixels within the training windows $\hat{R}_{t-1}$ and $\hat{R}_{t+1}$ can be expressed as

$$\hat{R}_{t-1} = AR_{t-2}A^T + BR_tB^T \tag{12}$$

$$\hat{R}_{t+1} = AR_tA^T + BR_{t+2}B^T. \tag{13}$$

Here, $A = \begin{pmatrix} C \\ D_A \\ E \end{pmatrix}$ and $B = \begin{pmatrix} C \\ D_B \\ E \end{pmatrix}$ with

$$C = \begin{bmatrix} 1 & 0 & .. & .. & .. & 0 \\ 0 & 1 & 0 & .. & .. & 0 \\ .. & .. & .. & .. & .. & .. \\ 0 & .. & 1 & 0 & .. & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & .. & .. & .. & 0 & 1 \\ 0 & .. & .. & 0 & 1 & 0 \\ .. & .. & .. & .. & .. & .. \\ 0 & .. & 0 & 1 & .. & 0 \end{bmatrix}$$

and

$$D_A = \begin{bmatrix} a_1 & a_2 & .. & a_{2L+1} & 0 & .. & .. & 0 \\ 0 & a_1 & a_2 & .. & a_{2L+1} & 0 & .. & 0 \\ .. & .. & .. & .. & .. & .. & .. & .. \\ 0 & .. & .. & 0 & a_1 & a_2 & .. & a_{2L+1} \end{bmatrix}$$

$$D_B = \begin{bmatrix} b_1 & b_2 & .. & b_{2L+1} & 0 & .. & .. & 0 \\ 0 & b_1 & b_2 & .. & b_{2L+1} & 0 & .. & 0 \\ .. & .. & .. & .. & .. & .. & .. & .. \\ 0 & .. & .. & 0 & b_1 & b_2 & .. & b_{2L+1} \end{bmatrix}$$

where C and E are $L \times (W_x \bullet W_y)$ matrices. D_A and D_B are $(W_x \bullet W_y - 2L) \times (W_x \bullet W_y)$ matrices.

The approximation of the actual pixels within the training window $\hat{R}_t$ can then be written as

$$\hat{R}_t = A\hat{R}_{t-1}A^T + B\hat{R}_{t+1}B^T. \tag{14}$$

Incorporating (12) and (13) into (14), $\hat{R}_t$ can be expressed as

$$\hat{R}_t = A\left(AR_{t-2}A^T + BR_tB^T\right)A^T + B\left(AR_tA^T + BR_{t+2}B^T\right)B^T. \tag{15}$$

Matrices A and B can be computed by assuming

$$\begin{aligned} R_t &= \hat{R}_t \\ &= A\left(AR_{t-2}A^T + BR_tB^T\right)A^T \\ &\quad + B\left(AR_tA^T + BR_{t+2}B^T\right). \end{aligned} \tag{16}$$

However, it will result in groups of multiple quadratic equations, which is a nonlinear optimization problem. If we take into account the spatial weight vector, the situation will be more complicated.

APPENDIX B

Define the spatio-temporal order to be L, and let the weight vector of the ith iteration be $\vec{W}^i = [W_p^i, W_f^i, W_s^i]^T$, with

$$W_p^i = \begin{bmatrix} W_p^i(-L,-L), W_p^i(-L,-L+1), \ldots, \\ W_p^i(-L,L), \\ W_p^i(-L+1,-L), \ldots, W_p^i(-L+1,L), \ldots, \\ W_p^i(L,L) \end{bmatrix}$$

$$W_f^i = \begin{bmatrix} W_f^i(-L,-L), W_f^i(-L,-L+1), \ldots, \\ W_f^i(-L,L), \\ W_f^i(-L+1,-L), \ldots, W_f^i(-L+1,L), \ldots, \\ W_f^i(L,L) \end{bmatrix}$$

and

$$W_s^i = \begin{bmatrix} W_s^i(-L,-L), W_s^i(-L,-L+1), \ldots, \\ W_s^i(-L,L), \\ W_s^i(-L+1,-L), \ldots, W_s^i(-L+1,L), \ldots, \\ W_s^i(0,-1) \end{bmatrix}.$$

According to the least square method, the equation to compute $\vec{W}^i$ can be formulated as

$$A^i\vec{W}^i = B^i \tag{17}$$

where A^i is a$\left(2\times(2\times L+1)^2 + \left(2L^2+2L\right)\right) \times \left(2\times(2\times L+1)^2 + \left(2L^2+2L\right)\right)$ matrix and B^i is a column vector with length $2\times(2\times L+1)^2 + \left(2L^2+2L\right)$. For presentation convenience, we set

$$A^i = \begin{bmatrix} A_{0,0}^i & A_{0,1}^i & A_{0,2}^i \\ A_{1,0}^i & A_{1,1}^i & A_{1,2}^i \\ A_{2,0}^i & A_{2,1}^i & A_{2,2}^i \end{bmatrix}$$

and

$$B^i = \begin{bmatrix} B_p^i & B_f^i & B_s^i \end{bmatrix}^T.$$

Here the size of sub-matrices $A_{0,0}^i$, $A_{0,1}^i$, $A_{1,0}^i$, and $A_{1,1}^i$ is $(2\times L+1)^2 \times (2\times L+1)^2$, the size of $A_{0,2}^i$ and $A_{1,2}^i$ is $\left(2L^2+2L\right)\times(2\times L+1)^2$. The size of $A_{2,0}^i$ and $A_{2,1}^i$ is $(2\times L+1)^2 \times \left(2L^2+2L\right)$ and the size of $A_{2,2}^i$ is $\left(2L^2+2L\right)\times\left(2L^2+2L\right)$. B_p^i and B_f^i are row vectors of length $(2\times L+1)^2$. B_s^i is a row vector with length $2L^2+2L$.

The element $a_{p,q}^i(m,n)$ of $A_{p,q}^i$, $0 \le p,q \le 1$, located at (m,n), can be computed as

$$\begin{aligned} a_{p,q}^i(m,n) = &\sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t-2\times(1-p)}(\tilde{m},\tilde{n}) \times R_{t-2\times(1-q)}(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-(-1)^p}^i(\tilde{m},\tilde{n}) \times \hat{R}_{t-(-1)^q}^i(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t+2\times p}(\tilde{m},\tilde{n}) \times R_{t+2\times q}(\tilde{m},\tilde{n}). \end{aligned} \tag{18a}$$

The element $a_{p,2}^i(m,n)$ of $A_{p,2}^i$, $0 \le p \le 1$, located at (m,n), can be computed as

$$\begin{aligned} a_{p,2}^i(m,n) = &\sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t-2\times(1-p)}(\tilde{m},\tilde{n}) \times \hat{R}_{t-1}^i(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-(-1)^p}^i(\tilde{m},\tilde{n}) \times R_t(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t+2\times p}(\tilde{m},\tilde{n}) \times \hat{R}_{t+1}^i(\tilde{m},\tilde{n}). \end{aligned} \tag{18b}$$

The element $a_{2,q}^i(m,n)$ of $A_{2,q}^i$, $0 \le q \le 1$, located at (m,n), can be computed as

$$\begin{aligned} a_{2,q}^i(m,n) = &\sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-1}^i(\tilde{m},\tilde{n}) \times R_{t-2\times(1-q)}(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_t(\tilde{m},\tilde{n}) \times \hat{R}_{t-(-1)^q}^i(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t+1}^i(\tilde{m},\tilde{n}) \times R_{t+2\times q}(\tilde{m},\tilde{n}). \end{aligned} \tag{18c}$$

The element $a_{2,2}^i(m,n)$ of $A_{2,2}^i$, located at (m,n) can be computed as

$$\begin{aligned} a_{2,2}^i(m,n) = &\sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-1}^i(\tilde{m},\tilde{n}) \times \hat{R}_{t-1}^i(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_t^i(\tilde{m},\tilde{n}) \times \hat{R}_t^i(\tilde{m},\tilde{n}) \\ &+ \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t+1}^i(\tilde{m},\tilde{n}) \times \hat{R}_{t+1}^i(\tilde{m},\tilde{n}) \end{aligned} \tag{18d}$$

with $\tilde{m} = k - L + m/(2\times L+1)$ and $\tilde{n} = k - L + n/(2\times L+1)$. The element located at $(0, m\times 2L+n)$ in

B_p^i and B_f^i can be computed as

$$b_r^i(0, m \times 2L + n) = \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t-2\times(1-r)}\left(\widehat{m}, \widehat{n}\right) \times \hat{R}_{t-1}^i(k,l) + \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-(-1)^r}^i\left(\widehat{m}, \widehat{n}\right) \times R_t(k,l) + \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_{t+2\times r}\left(\widehat{m}, \widehat{n}\right) \times \hat{R}_{t+1}^i(k,l). \tag{19a}$$

Here, when r is set to be 0, it represents the element of B_p^i and when r is set to be 1, it represents the element of B_f^i.

The element located at $(0, m \times 2L + n)$ in B_s^i can be computed as

$$b_r^i(0, m \times 2L + n) = \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t-1}^i\left(\widehat{m}, \widehat{n}\right) \times \hat{R}_{t-1}^i(k,l) + \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} R_t\left(\widehat{m}, \widehat{n}\right) \times R_t(k,l) + \sum_{k=0}^{W_x}\sum_{l=0}^{W_y} \hat{R}_{t+1}^i\left(\widehat{m}, \widehat{n}\right) \times \hat{R}_{t+1}^i(k,l) \tag{19b}$$

with $\widehat{m} = k - (2L+1) + m$ and $\widehat{n} = k - (2L+1) + n$.

REFERENCES

[1] G. Dane and T. Q. Nguyen, "Optimal temporal interpolation filter for motion-compensated frame rate upconversion," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 978–991, Apr. 2006.

[2] Y. Yang, Y. Shin, and J. Wu, "Quality enhancement of frame rate upconverted video by adaptive frame skip and reliable motion extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 12, pp. 1700–1713, Dec. 2007.

[3] R. Castagno, P. Haavisto, and G. Ramponi, "A method for motion adaptive frame rate upconversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 5, pp. 436–446, Oct. 1996.

[4] S.-H. Lee, Y.-C. Shin, S.-J. Yang, H.-H. Moon, and R.-H. Park, "Adaptive MC interpolation for frame rate upconversion," *IEEE Trans. Consumer Electron.*, vol. 48, no. 3, pp. 444–450, Aug. 2002.

[5] B. T. Choi, S. H. Lee, and S. J. Ko, "New frame rate upconversion using bi-directional motion estimation," *IEEE Trans. Consumer Electron.*, vol. 46, no. 3, pp. 603–609, Aug. 2000.

[6] G. I. Lee, B. W. Jeon, R. H. Park, and S. H. Lee, "Hierarchical motion-compensated frame rate upconversion based on the Gaussian/Laplacian pyramid," in *Proc. IEEE Int. Conf. Consumer Electron.*, 2003, pp. 350–351.

[7] Z. Gan, L. Qi, and X. Zhu, "Motion-compensated frame interpolation based on H.264 decoder," *Electron. Lett.*, vol. 43, no. 2, pp. 96–98, Jan. 2007.

[8] G. Dane and T. Nguyen, "Motion vector processing for frame rate upconversion," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 2004, pp. 309–312.

[9] G. de Haan, P. W. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[10] P. Csillag and L. Borocszky, "Enhancement of video data using MC postprocessing techniques," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 1997, pp. 2897–2900.

[11] D. W. Kim, J. T. Kim, and I. H. Ra, "A new video interpolation technique based on motion-adaptive subsampling," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 782–786, Aug. 1999.

[12] S. Liu, Z. Yan, J. W. Kim, and C. C. Jay Kuo, "Global/local motion-compensated frame interpolation for low-bit-rate video," in *Proc. SPIE Image Visual Commun. Process. 2000*, San Jose, CA, Jan. 2000, pp. 223–234.

[13] B. D. Choi, J. W. Han, C. S. Kim, and S. J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

[14] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion-compensated frame interpolation method," in *Proc. ISCAS*, vol. 5. May. 2005, pp. 4927–4930.

[15] K. Hilman, H. Park, and Y. Kim, "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 869–877, Sep. 2000.

[16] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2002, pp. 313–316.

[17] T. Kuo and C.-C. J. Kuo, "Motion-compensated interpolation for low-bit-rate video quality enhancement," in *Proc. SPIE Int. Symp. Optical Sci., Eng. Instrumentation*, May 1998.

[18] S. Lee, O. Kwon, and R. Park, "Weighted-adaptive motion-compensated frame rate-upconversion," *IEEE Trans. Consumer Electron.*, vol. 49, no. 3, pp. 485–492, Aug. 2003.

[19] R. Krishnamurthy, J. W. Woods, and P. Moulin, "Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 713–726, Aug. 1999.

[20] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[21] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

[22] X. Wu, K. U. Barthel, and W. Zhang, "Piecewise 2-D autoregression for predictive image coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 1998, pp. 901–904.

[23] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner, "Detection of missing data in image sequences," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1496–1508, Nov. 1995.

[24] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner, "Interpolation of missing data in image sequences," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1509–1519, Nov. 1995.

[25] S. N. Efstratiadis and A. K. Katsaggelos, "A model-based pel-recursive motion estimation algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, 1990, pp. 1973–1976.

[26] X. Zhang and X. Wu, "Edge-guided perceptual image coding via adaptive interpolation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2007, pp. 1459–1462.

[27] D. Pokrajac, R. L. Hoskinson, and Z. Obradovic, "Modeling spatial-temporal data with a short observation history," *Knowl. Inf. Syst.*, vol. 5, no. 3, pp. 368–386, Sep. 2003.

[28] X. Li. (2006). "Least-square based prediction for backward adaptive video coding," *EURASIP J. Appl. Signal Process., Special Issue H.264 Beyond* [Online]. Available: http://www.hindawi.com/GetArticle.aspx?doi=10.155/ASP/2006/90542

[29] Y. Zhang, D. Zhao, X. Ji, R. Wang, and X. Chen, "A spatial-temporal autoregressive frame rate upconversion," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2007, pp. 441–444.

Yongbing Zhang received the B.A. degree in english and the M.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2004 and 2006, respectively. He is currently pursuing the Ph.D. degree at the Department of Computer Science, Harbin Institute of Technology.

His research interests include video processing, image and video coding, video streaming, and transmission.

Debin Zhao received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1985, 1988, and 1998, respectively.

He was a Research Fellow in the Department of Computer Science, City University of Hong Kong, from 1989 to 1993. He is now with the Department of Computer Science, Harbin Institute of Technology and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include data compression, image processing, and human–machine interface. He has published two books and over 200 scientific papers.

Xiangyang Ji received the B.S. and the M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences and the Graduate School of Chinese Academy of Science, Beijing, China, in 2008.

He is now with Broadcast Networks and Digital Media Laboratory, Automation Department, Tsinghua University, Beijing, China. He has authored or co-authored over 40 conference and journal papers. His research interests include video/image coding, video streaming, and multimedia processing.

Ronggang Wang received the B.S. degree in information science from Hei Long Jiang University in 1998, the M.S. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 2002, and the Ph.D. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Sciences, in 2006.

Since 2006, he has been a Researcher with France Telecom Research and Development, Beijing Company Limited, China. His research interests are video compression, video streaming and multimedia processing.

Wen Gao (F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan.

He is a Professor of computer science with Peking University, Beijing, China. Before joining Peking University, he was a full Professor of computer science at the Harbin Institute of Technology from 1991 to 1995. At the Chinese Academy of Sciences (CAS), he served from 1996 to 2005 as a Professor, the Managing Director of the Institute of Computing Technology, CAS, the Executive Vice President of the Graduate School of CAS, and a Vice President of the University of Science and Technology of China. He has published extensively, including four books and over 500 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics.

Dr. Gao is the Editor-in-Chief of the *Journal of Computer* (a journal of the Chinese Computer Federation), an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA, an Associate Editor of IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, an area editor of *EURASIP Journal of Image Communications*, and an editor of the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, and also served on the advisory and technical committees of numerous professional organizations.