A SPATIO-TEMPORAL AUTOREGRESSIVE FRAME RATE UP CONVERSION SCHEME

Yongbing Zhang[1], Debin Zhao[1], Xiangyang Ji[2], Ronggang Wang[3], and Xilin Chen[2]

[1]Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P.R. China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing10080, P.R. China
[3]France Telecom R&D Beijing Co., Ltd
{ybzhang, xyji, dbzhao, xlchen}@jdl.ac.cn, ronggang.wang@orange-ftgroup.com

ABSTRACT

A spatio-temporal autoregressive model is proposed in this paper to address the problem of frame rate up conversion. Every pixel in a skipped frame is generated as a linear combination of pixel values from forward and backward reference frames. At the beginning of the presented scheme, the coarse model parameters are computed according to the given initial pixel values for skipped frames. Then the coarse parameters are refined by an iteration process, during which we also interpolate the original low rate frames by the two closest generated skipped frames to derive more accurate parameters. Experimental results verify that the proposed algorithm significantly improves both the subjective and objective quality of the interpolated frames.

***Index Terms*—**Frame rate up conversion, Spatio-temporal autoregressive, Linear combination

1. INTRODUCTION

Frame rate up conversion (FRUC), which is also referred as picture rate conversion or temporal frame interpolation in the literature, is the conversion from lower frame rates to higher ones. Besides the necessity for conversion among various display formats with different frame rates [1], FRUC can also be used in low bit rate video coding [2]. With low bandwidth limits, one can send the full frame rate video at the cost of introducing annoying artifacts. Alternatively, the frame rate may be reduced by half so that each frame has better quality. In the latter case, a recovery mechanism utilizing FRUC is necessary at the decoder to display the video at a full frame rate.

Generally, FRUC can be divided into two categories. The first category, such as frame repetition and linear interpolation (LI), interpolates the skipped frame along temporal axis without taking the object motion into account. However, these algorithms produce “jerkiness” into the motion portrayal and blurring on object boundaries, respectively [3]. The second category interpolates the skipped frame along motion trajectory exploiting motion information between successive frames [4], which are also referred as motion-compensated interpolation (MCI). Given true and accurate motion vectors, MCI outperforms the first strategy. The interpolator that predicts the skipped frame can be either linear (Haar filter), or nonlinear (median operation [5]). Yet, as one interpolates the additional frames to increase the frame rate of the video, local artifacts are introduced due to the incorrect estimation of motion vectors. For the MCI case, motion may look jerky, not smooth or very choppy. To get higher visual quality frames, we need to use true motion vector fields, whereas block matching does poorly with this, especially for the regions across object boundaries and when the moving object is too small. Although block artifacts can be reduced by MCI combined with overlapped block motion compensation (OBMC) [6], which is also referred as MCI-OBMC, sometimes the generated frames still look unpleasant. What’s more, traditional FRUC algorithm can’t cope with discontinuity in the velocity plane.

In order to overcome the shortcomings of MCI as well as MCI-OBMC and achieve higher visual quality frames, we propose a STAR-FRUC scheme in this paper. STAR was proposed for forecasting both synthetic and real-life data using a short observation history in [7]. In the proposed algorithm, STAR is introduced to approximate the skipped frames by suitable interpolation using forward and backward reference frames. There are two main contributions in this work. Firstly, in contrast to previous FRUC algorithms, here a generated pixel depends on the values from all neighboring locations on a rectangular lattice within a pre-specified distance in neighboring frames. Secondly, the STAR parameters are derived by a joint optimization process.

In the rest of the paper, we first present the proposed STAR-FRUC in detail in Section 2. Then the effectiveness of the proposed algorithm is demonstrated by experimental results in Section 3. Finally, the paper is concluded in Section 4.

2. SPATIO-TEMPORAL AUTOREGRESSIVE MODEL

A STAR model is defined on a uniform three-dimensional cubic grid determined by sampling distances Δs in both spatial directions and Δt along temporal axis. Model samples are taken at time instances $k\zeta$ where $k \in \{1,2,\ldots\}$ and ζ is a time sampling interval for the up converted frame rate.

For each skipped frame in time instance t, the interpolated pixel value $\hat{p}_t(m,n)$ at spatial location (m, n) is generated as a linear combination of pixel values taken from the closest forward and backward low rate reference frames at the same location and its spatial neighbors. The spatial neighborhood is specified as a (2L+1) × (2L+1) square bounded by locations (m ± L, n ± L). Parameter L will be referred as spatial order of a STAR model (for L=2, the STAR model is illustrated in Fig. 1).

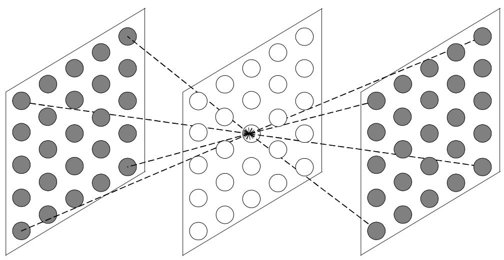

Fig. 1. STAR model on a uniform grid with spatial order L=2

We express the value of $\hat{p}_t(m,n)$ as

$$\hat{p}_t(m,n) = \sum_{i=1}^{2}\sum_{k=-L}^{L}\sum_{l=-L}^{L} p_{t+(-1)^i}(m+k,n+l) \bullet \phi_{t+(-1)^i}(k,l) \qquad (1)$$

where $\phi_{t-1}(k,l)$ and $\phi_{t+1}(k,l)$ denote the model coefficients for pixel $\hat{p}_t(m,n)$ in the forward and backward reference frames, respectively.

Based on the assumption that current block is undergoing translational movement, MCI algorithm gives each block a unique motion vector. Nevertheless, this assumption does not hold true in object boundaries or complex motion regions. Fortunately, STAR can tackle these problems quite well, as it can adjust its weights for different areas to make the interpolated frames more accurate and smooth. Furthermore, OBMC can be seen as a specific case of STAR. That is because if we set certain parameters of STAR to be specific values, while other parameters are set to be zero, STAR equals to OBMC. It can be seen that STAR parameter matrices $\Phi_{forward}$ and $\Phi_{backward}$ play crucial roles for the values of interpolated pixels in the generated frames, where $\Phi_{forward} = \{\phi_{t-1}(k,l)\}$ and $\Phi_{backward} = \{\phi_{t+1}(k,l)\}$, respectively, and $k, l \in \{-L, -L+1, \ldots, L\}$. Details of parameter estimation will be described in the following subsection.

2.1 Parameter estimation

According to the characteristic of pixel-wise stationary, which means the model parameters change slowly, we assume the STAR model parameters remain the same in the nearby frames within a region. In the proposed scheme, we divide each frame into non-overlapped 16×16 macroblocks (MB) and process the parameter estimation in MB-wise. Although the available data are only the reconstructed low rate frames, more accurate STAR parameters can be derived by interpolating the skipped frames using reconstructed low rate frames as well as interpolating the reconstructed frames using generated frames. The process is shown in Fig. 2.

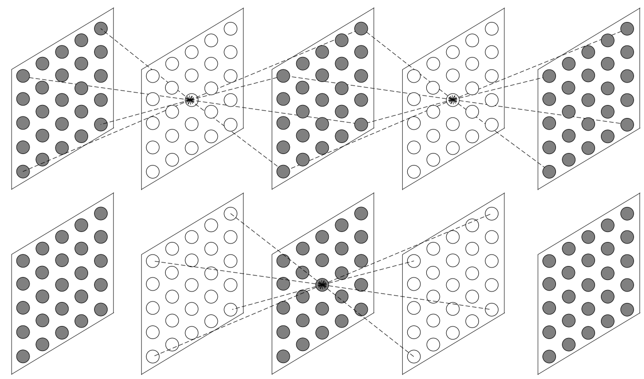

Fig. 2. Interpolation of the skipped frames and the interpolation of the reconstructed frames by generated frames

As is depicted in Fig. 2, we set every 5 successive MBs (including reconstructed and generated MBs) as a group of MBs (GOMB) and assume STAR parameters remain the same in each GOMB. In Fig. 2, the frames filled with solid circles are reconstructed frames and the frames filled with hollow circles are generated ones. Skipped frames t-1 and t+1 are interpolated by their closest forward and backward reference frames at first (as shown in the top of Fig. 2), and then reconstructed frame t is interpolated by the nearest generated frames (as shown in the bottom of Fig. 2). Interpolation of frames t-1 and t+1 utilizes reconstructed low rate frame t, rather than the frame interpolated by frames t-1 and t+1. The aim of interpolating reconstructed frame t is to derive more accurate STAR parameters. Pixel values of the skipped frames are interpolated according to Eq. (2).

$$\begin{aligned}\hat{p}^i_{t+(-1)^j}(m,n) = &\sum_{k=-L}^{L}\sum_{l=-L}^{L} p_{t-2*(j\%2)}(m+k,n+l) \bullet \phi^{i-1}_{forward}(k,l) \\ &+ \sum_{u=-L}^{L}\sum_{v=-L}^{L} p_{t+2*(1-j\%2)}(m+u,n+v) \bullet \phi^{i-1}_{backward}(u,v)\end{aligned} \qquad (2)$$

Here, j is set to be 1 and 2 respectively, superscript i represents ith iteration process, subindex t represents time instance. $\hat{p}^i_t(m,n)$ represents the interpolated pixel value in the ith iteration. $p_t(m,n)$ represents the reconstructed low frame rate pixel value located at (m, n). $\phi^{i}_{forward}(k,l)$ and $\phi^{i}_{backward}(u,v)$, where $k,l \in \{-L, -L+1, \cdots, L\}$ and $u,v \in \{-L, -L+1, \cdots, L\}$, represent the forward and backward STAR weighting parameters, respectively. Reconstructed frame t is interpolated using the generated frames t-1 and t+1 as Eq. (3)

$$\begin{aligned}\hat{p}^{i}_{t}(m,n) = &\sum_{k=-L}^{L}\sum_{l=-L}^{L}\hat{p}^{i}_{t-1}(m+k,n+l)\bullet\phi^{i-1}_{forward}(k,l)\\ &+\sum_{u=-L}^{L}\sum_{v=-L}^{L}\hat{p}^{i}_{t+1}(m+u,n+v)\bullet\phi^{i-1}_{backward}(u,v)\end{aligned} \tag{3}$$

Combining with Eq. (2), and Eq. (3), STAR parameters $\phi_{forward}$ and $\phi_{backward}$ in the ith iteration are computed according to Eq. (4).

$$\Phi_{i} = \arg\min_{\phi_{forward},\phi_{backward}}\left\{\begin{aligned}&\sum\sum_{(m,n)\in S}\left\|\hat{p}^{i}_{t-1}(m,n)-\hat{p}^{i-1}_{t-1}(m,n)\right\|^{2}\\ &+\sum\sum_{(m,n)\in S}\left\|\hat{p}^{i}_{t}(m,n)-p_{t}(m,n)\right\|^{2}\\ &+\sum\sum_{(m,n)\in S}\left\|\hat{p}^{i}_{t+1}(m,n)-\hat{p}^{i-1}_{t+1}(m,n)\right\|^{2}\end{aligned}\right\} \tag{4}$$

The proposed parameter estimation algorithm is implemented as follows:

Step 1: Set iteration times i to be 0. For each (m, n), $\hat{p}^{0}_{t-1}(m,n)$ and $\hat{p}^{0}_{t+1}(m,n)$ are set to be the initial values for the skipped frames t-1 and t+1. For instance, MCI results could be set to be the initial values.

Step 2: Combining Eq. (2) to Eq. (4), coarse STAR parameters are computed.

Step 3: i = i+1. Compute $\hat{p}^{i}_{t-1}(m,n)$, $\hat{p}^{i}_{t+1}(m,n)$ and $\hat{p}^{i}_{t}(m,n)$ according to Eq. (2), and Eq. (3), respectively.

Step 4: Compute STAR parameters for the ith iteration according to Eq. (4), and then compute the distortion according to the right part of Eq. (4). Then compute the difference of the distortion between the current and previous iteration.

Step 5: Repeat *Step 3* and *Step 4*, until the difference of distortion in *Step 4* is below a threshold or the maximum iteration time is reached.

2.2 Parameter selection

In order to derive more accurate and robust STAR parameters, the GOMB window moves 2 frames backward after one skipped frame is interpolated. Consequently, another problem arises, that is one frame may belong to two GOMBs, as shown in Fig. 3, where the frames indicated by dotted lines are the skipped frames, whereas the others are the original ones. From Fig. 3, we can see that the skipped frame t+1 belongs to the *nth* GOMB, while at the same time it also belongs to the n+1 *th* GOMB.

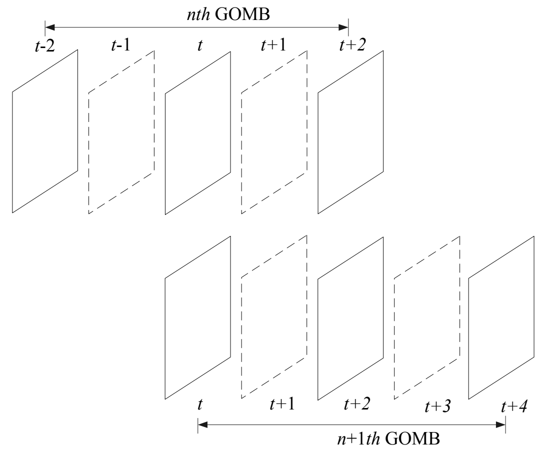

Fig. 3 The relation between each frame and its GOMB

For each of the two succesive GOMBs to which frame t+1 belongs, we introduce

$$distortion = \sum\sum_{(m,n)\in S}\left\{\begin{aligned}&\left\|\hat{p}^{optimal}_{t-1}(m,n)-\hat{p}^{optimal\_prev}_{t-1}(m,n)\right\|^{2}\\ &+\left\|\hat{p}^{optimal}_{t}(m,n)-p_{t}(m,n)\right\|^{2}\\ &+\left\|\hat{p}^{optimal}_{t+1}(m,n)-\hat{p}^{optimal\_prev}_{t+1}(m,n)\right\|^{2}\end{aligned}\right\} \tag{5}$$

where $\hat{p}^{optimal}_{t}(m,n)$ means the interpolated pixel value when STAR parameters are converged or the maximum iteration time is reached, and $\hat{p}^{optimal\_prev}_{t}(m,n)$ is the interpolated pixel value in the iteration before STAR parameters are converged or the maximum iteration time is reached. Between the *nth* and the n+1*th* GOMB, we choose the parameters whose distortion is smaller to interpolate the skipped frame t. For instance, if the distortion within the *nth* GOMB is smaller than that in the n+1*th* GOMB, we use the parameters of the *nth* GOMB; otherwise, we choose the parameters of the n+1*th* GOMB.

3. EXPERIMENTAL RESULTS

In order to verify the validity of the proposed algorithm, various 15 frames/s sequences reconstructed from an H.264 decoder are evaluated by STAR-FRUC to generate 30 frames/s sequences. The sequences are encoded as IBPBP structure and the QP is set to be 30. Every other frame is skipped during the encoding and is interpolated using STAR-FRUC. In the experiment, L is set to be 3 and the maximum number of iterations is set to be 5. The proposed method is compared with LI, MCI, and MCI-OBMC. For the MCI case, full search block matching algorithm for motion estimation (ME) is applied. For each skipped frame, ME is performed in not only the forward but also the backward reference frames, and the motion vectors leading to smaller distortions are used to perform MCI. The MCI results are then used as initial values for STAR-FRUC. Peak signal noise ratios (PSNR) values, which are computed compared to the corresponding original frames, are averaged over 50 skipped frames and they are given in Table 1.

Table 1. Average PSNRs (dB) of interpolated frames generated by different algorithms

Sequence	LI	MCI	MCI-OBMC	STAR-FRUC
Mobile (QCIF)	29.98	31.03	31.08	31.82
Paris (CIF)	29.04	31.12	31.39	31.67
Tempete (CIF)	28.86	29.45	29.55	29.95
Foreman (CIF)	29.96	32.62	32.79	33.04
Spincalendar (HD)	26.34	28.88	29.28	29.70

The first column represents the PSNR value of LI, where pixels in skipped frames are computed by simply averaging the collocated pixel values of the forward and backward reference frames. The subsequent columns are MCI, MCI-OBMC, and the proposed STAR-FRUC, respectively. From Table 1, we can see that STAR-FRUC outperforms LI, MCI, and MCI-OBMC for the entire test sequences. MCI outperforms LI greatly, due to the adoption of motion compensation. MCI-OBMC achieves superior performance compared to MCI. Nevertheless, the performance enhancement is rather poor, especially for *Mobile* which gains only 0.05 dB compared to MCI. STAR-FRUC, as a more general case of MCI-OBMC, achieves the highest performance.

Fig. 4 depicts the visual comparison of the 28th skipped frame in *Foreman*. Using LI without considering motion between frames results in the worst quality. MCI, exploiting motion displacement existed in different frames, achieves better visual quality compared to LI. However there are significant artifacts in the mouth area. MCI-OBMC achieves higher PSNR values compared to MCI. However, there are still some artifacts in the mouth area. The proposed STAR algorithm provides the best visual quality and highest PSNR values among the four FRUC methods, due to the full exploitation of similarities within a specified spatial and temporal neighborhood.

4. CONCLUSIONS

In this paper, a novel STAR-FRUC scheme is proposed for frame rate up conversion. It exploits the spatio-temporal interactions among pixel values within successive video frames. The most remarkable difference between traditional FRUC and the proposed STAR-FRUC is that each pixel is interpolated as a linear combination of the pixels within its pre-specified spatial and temporal neighborhood.

(i) (ii) (iii) (iv)

Fig. 4: Visual comparison of FRUC methods: (i) LI-29.957dB, (ii) MCI-32.619dB, (iii) MCI-OBMC-32.786dB, (iv) STAR-33.04dB

Evaluation of the proposed method is carried on sequences encoded and decoded by H.264. Although the complexity of STAR-FRUC is comparatively higher than other FRUC algorithms, the experimental results confirm that the proposed STAR-FRUC outperforms LI, MCI and MCI-OBMC algorithms for all the tested sequences both subjectively and objectively. Future works may include some optimization techniques to decrease the complexity of the proposed STAR-FRUC.

5. REFERENCES

[1] S.Fujiwara and Taguchi, A, "Motion-compensated frame rate up-conversion based on block matching algorithm with multi size blocks," in Proc. International Symposium On Intelligent Signal Processing and Communication Systems, 2005, pp. 353-356.

[2] G. Dane and Truong Nguyen., "Quality enhancement in standard compliant frame rate up conversion by motion smoothing", in Proc. EUSIPCO, 2005

[3] K.A. Bugwadia, E.D. Petajan, and N.N. Puri, "Progressive-scan rate up-conversion of 24/30 source materials for HDTV," *IEEE Trans. Consum. Electron*. vol. 42, no. 3, pp. 312-321, Aug. 1996

[4] S.C. Han and J.W.Woods, "Frame-rate up-conversion using transmitted motion and segmentation fields for very low bit-rate video coding," in *Proc. Int. Conf. Image Processing,* vol. 1, pp.747-750, Oct. 1997

[5]O.A.Ojo, G de Haan, " Robust motion-compensated up conversion," *IEEE Trans. Consum. Electron*., vol. 43, no. 4, pp. 1045-1056, Nov, 1997

[6] B.T. Choi, S.H. Lee, and S.J. Ko, "New frame rate up-conversion using bi-directional motion estimation," *IEEE Trans. Consum. Electron*, vol. 46, no. 4, pp. 603-609, Aug. 2000

[7] D. Pokrajac, R.L.Hoskinson and Z.Obradovic, "Modeling spatial-temporal data with a short observation history," *Knowl. Inf. Syst.* 5(3): 368-386 (2003).