A 3D Auto-Regressive Model for Bi-Directional Prediction

Yongbing Zhang*[a], Debin Zhao[a], Siwei Ma[b], Ronggang Wang[c], Wen Gao[b]
[a]Department of Computer Science, Harbin Institute of Technology, Harbin, 150001, China
[b]Institute of Digital Media, Peking University, Beijing, 100871, China
[c]France Telecom R&D, Beijing, Co., Ltd
e-mail: ybzhang@jdl.ac.cn

ABSTRACT

In this paper, a 3D auto-regressive (AR) model is proposed for bi-directional prediction. The prediction is composed of two 3D AR models, which are along the forward and backward directions, respectively. Applying the 3D AR model, each pixel in the current frame is predicted as a weighted summation of pixels within a spatial neighborhood along the forward/backward motions within the forward/backward reference frames. Ultimately, the prediction of each pixel is obtained as the combination of predictions generated by the two 3D AR models. To derive accurate AR coefficients, this paper proposes a framework that performs simultaneous coefficient estimation and image interpolation. As opposed to other methods, the predicted pixels generated by one 3D AR model are further used to predict the pixels in adjacent frame along the motion trajectory. Consequently, each pixel in one forward/backward reference frame can be predicted as a nonlinear combination of pixels within an enlarged spatial neighborhood along the motion in one backward/forward reference frame. An iterative algorithm using a nonlinear least squares method is then devised to compute the optimum 3D AR coefficients. Various experiments conducted in this paper have confirmed that the proposed method has superior performance for bi-directional prediction.

Keywords: 3D AR, bi-directional prediction, nonlinear least squares

1. INTRODUCTION

Motion compensated prediction (MCP) has a broad application in video compression, since it removes the temporal redundancy by estimating the motion vector field between successive frames [1]. MCP has been adopted in various video coding standards such as ITU-T Recommendations H. 261 [2] H.263 [3] [4], and H.264/AVC [5] [6]. However, MCP does not achieve good results in cases there are intensity variations between successive frames or the object shape is not consistent with the block shape. That's due to the reason that MCP is based on the assumption that the object is performing translational motion between adjacent images, which is not always hold true for natural videos. By extending traditional MCP, overlapped block motion compensation (OBMC) [7] [8] was proposed to further increase the performance of linear prediction. OBMC is based on multi-hypothesis and predicts one block as the weighted summation of multi blocks, pointed by multiple motion vectors, and thus can reduce the inefficiency brought by MCP when the block shape is not consistent with the object shape. However, constant predictor coefficients are used to combine linearly hypothesis of a multi-hypothesis in OBMC. Consequently, it may sometimes oversmooth the edges of the image and degrade the visual quality of the predicted image.

Auto-regressive (AR) model [9], which is an efficient, compact description of random process, is able to have desirable performance for the linear prediction. AR models have attracted a lot attention of image/video applications. Both spatial AR predictor [9] and temporal AR predictors [10] [11] have been proposed to improve the prediction accuracy. In [9], predictive pixels were obtained by observed "prior" block of pixels in a block raster scan of the current image. In [10], geometric duality was applied to obtain a high resolution sequence based on the low resolution one, and in [11] each pixel was predicted as a linear combination of pixels in spatio-temporal neighborhood, and then the resulting error was quantized and transmitted to decoder. In this method, the motion information is embedded by the AR coefficients and the AR coefficients have to be transmitted to the decoder the match the encoder lossless.

Inspired by the superior property of the AR model, in this paper a 3D AR model was proposed to predict the current frame based on the observations in the previous and following frames. In the 3D AR model, each pixel in the current frame is predicted as a weighted summation of pixels within a spatial neighborhood along the motion trajectory in the forward/backward reference frame. To avoid transmitting the AR coefficients so as to save the bandwidth in video coding, we first predict each pixel in the current block as a weighted summation of corresponding pixels in the forward/backward reference frames. Then we propagate the AR interpolation along the motion trajectory, which means that we use the predicted pixels in the current block to further predict the pixels in the adjacent backward/forward reference frames along the motion trajectory. Consequently, each pixel in the forward/backward reference frame can be predicted as a nonlinear combination of pixels within an enlarged spatial neighborhood along the motion in the backward/forward reference frame. To tackle such a nonlinear problem, an iterative prediction algorithm using a nonlinear least squares method is then devised to compute the optimum AR coefficients. Ultimately, the prediction of each pixel is obtained as the combination of predictions generated by the forward and backward 3D AR models.

The organization of this paper is as follows. Section 2 describes the AR model for linear prediction, followed by an iterative algorithm for deriving AR coefficients in Section 3. Experimental results are presented in Section 4. And finally, this paper is concluded in Section 5.

2. AUTO-REGRESSIVE MODEL FOR LINEAR PREDICTION

The proposed AR model aims to predict the picture data based on the previous and following observation history. Let $\mathbf{Y}_{2k+1}$ be a rectangular block of pixels in the current frame $2k+1$, $\mathbf{X}_{2k}$ be the similar block in the previous frame $2k$ along the forward motion trajectory and $\mathbf{X}_{2k+2}$ be the similar block in the following frame $2k+2$ along the backward motion trajectory. $\mathbf{Y}_{2k+1}$ can be predicted by the proposed linear AR model based on pixels in $\mathbf{X}_{2k}$ (forward AR model) as well as pixels in $\mathbf{X}_{2k+2}$ (backward AR model).

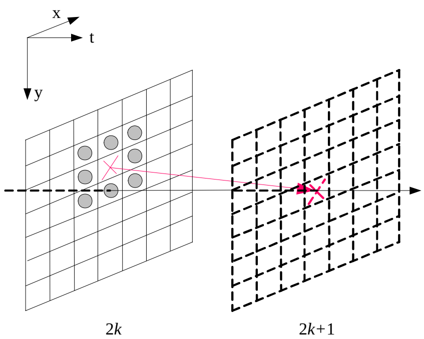

Fig. 1 the forward linear AR model

The forward linear AR model is depicted in Fig. 1, where for each current pixel in the block $\mathbf{Y}_{2k+1}$, as indicated by the dashed cross, we first find its corresponding pixel within $\mathbf{X}_{2k}$, as indicated by the solid cross, pointed by the forward motion vector. And then we approximate the current pixel as a linear combination of pixels within a spatial neighborhood, centered on the corresponding pixel, as indicated by the solid cross in Fig. 1, in the forward reference frame $2k$. Due to the piecewise characteristics of natural image, we assume the AR coefficients remain the same for all the pixels within current block $\mathbf{Y}_{2k+1}$. Similarly, the backward linear AR model, which is shown in Fig. 2, is applied in the backward reference frame $2k+2$ along the backward motion vector.

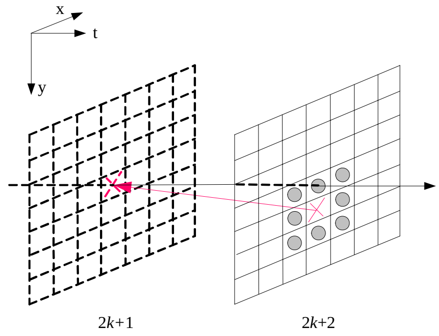

Fig. 2 the backward linear AR model

Since the forward and backward 3D AR models are quite similar, we will take the forward AR model for example in the following. Denote $X_{2k}(m,n)$ as the pixel value located at (m,n) within frame $2k$, according to the forward AR model, the predicted pixel in block $\mathbf{Y}_{2k+1}$ can be approximated as

$$\hat{Y}_{2k+1}(m,n) = \sum_{-l \le (i,j) \le l} X_{2k}\left(i+m+\lfloor v_{fx} \rfloor, j+n+\lfloor v_{fy} \rfloor\right) \bullet \alpha_{i,j} + n_{2k+1}(m,n), \tag{1}$$

where (v_{fx}, v_{fy}) represents the forward motion with fractional pixel accuracy from frame $2k+1$ to $2k$, $\lfloor \ldots \rfloor$ is the floor function, which maps v_{fx} and v_{fy} to the full pixel position, $\alpha_{i,j}$ is the forward AR coefficient located at (i,j), and n_{2k+1} is the additive Gaussian white noise. Here l is defined to be radius of the AR filter, and thus the size of the AR filter is $(2l+1)\times(2l+1)$.

If we assume the current block to be predicted is of size $W \times H$, and arrange the pixels in the current block as an $W \times H$ vector $\hat{\mathbf{Y}}_{2k+1} = \left(\hat{Y}_{2k+1}(0,0), \hat{Y}_{2k+1}(0,1), ..\hat{Y}_{2k+1}(W-1, H-1)\right)^t$, representing the concatenated and lexicographically ordered intensity values in the predicted block within frame $2k+1$, then equation (1) can be rewritten as

$$\hat{\mathbf{Y}}_{2k+1} = f\left(\mathbf{X}_{2k}\right)\boldsymbol{\alpha} + \mathbf{n}_{2k+1}, \tag{2}$$

where $\mathbf{X}_{2k} = \left(X_{2k}(0,0), X_{2k}(0,1), ..X_{2k}(W-1, H-1)\right)^t$ is a $W \times H$ dimensional vector, representing the concatenated and lexicographically ordered intensity values in the block, pointed by the forward motion vector within frame $2k$, $f\left(\mathbf{X}_{2k}\right)$ is a function which transfers $\mathbf{X}_{2k}$ to a $(W \bullet H)\times\left((2L+1)\bullet(2L+1)\right)$ dimensional matrix, $\boldsymbol{\alpha} = \left(a_{-L,-L}, a_{-L,-L+1}, .., a_{L,L}\right)^t$ is the forward AR coefficient vector, $\mathbf{n}_{2k+1}$ is the additive Gaussian white noise vector. Here each row vector $f_m\left(\mathbf{X}_{2k}\right)$, $m = 0,1,\ldots,W \times H - 1$, corresponds to the 3D AR spatial neighborhood, in a concatenated and lexicographically order, of the pixel m. The forward 3D AR coefficient vector $\boldsymbol{\alpha}$ should be chosen to be the "best" in some sense. Here we will use the most common measure of performance of a predictor: the mean squared error (MSE). Define the resulting MSE as

$$\begin{aligned} \varepsilon^2\left(\hat{\mathbf{Y}}_{2k+1}\right) &= E\left(\left\|\mathbf{Y}_{2k+1} - \hat{\mathbf{Y}}_{2k+1}\right\|^2\right) \equiv \mathrm{E}\left[\frac{1}{2}\left(\mathbf{Y}_{2k+1} - \hat{\mathbf{Y}}_{2k+1}\right)^t \left(\mathbf{Y}_{2k+1} - \hat{\mathbf{Y}}_{2k+1}\right)\right] \\ &= \sum_{m=0}^{W\times H-1} E\left[\left(Y_{2k+1}(m) - \hat{Y}_{2k+1}(m)\right)^2\right] \end{aligned}. \tag{3}$$

The MSE as a performance criterion can be viewed as a measure of how much the energy of the signal is reduced by removing the predictable information based on the observables from it. Since the goal of a predictor is to remove this predictable information, a better predictor corresponds to a smaller MSE. However, this method does not work for the case when $\mathbf{Y}_{2k+1}$ is not available, i. e. at decoder side if no coefficients are written in the bit stream. To tackle such a problem, an iterative algorithm for deriving more reliable AR coefficient is proposed in the following section.

3. AN ITERATIVE ALGORITHM FOR DERIVING AR COEFFICIENT

Based on the AR model mentioned in Section 2, we continue the approximation of pixels in the temporal axis along the motion trajectory. That is to say, assume we have obtained the predicted pixel values in frame $2k+1$, and then we use the same 3D AR coefficients to approximate the pixels within frame $2k+2$ along the motion trajectory. And this process is illustrated in Fig. 3.

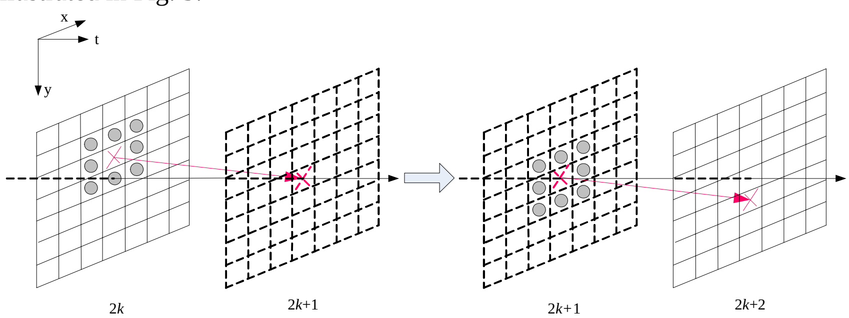

Fig. 3 the propagation of 3D AR process in temporal axis along the motion trajectory

The assumption that the 3D AR coefficients, used to approximate pixels within frame $2k+2$, remain the same as those used to approximate the pixels within frame $2k+1$ is quite reasonable since there is a high redundancy between pixels along the motion trajectory from frame $2k$ to $2k+2$. Based on such an assumption, pixels in frame $2k+2$ can be approximated as

$$\hat{\mathbf{X}}_{2k+2} = f\left(\hat{\mathbf{Y}}_{2k+1}\right)\boldsymbol{\alpha} + \mathbf{n}_{2k+2}, \tag{4}$$

where $\hat{\mathbf{X}}_{2k+2}$ is a $W \times H$ dimensional vector, representing the concatenated and lexicographically ordered value, which corresponds to pixels within the block pointed by the backward motion vector in frame $2k+2$, $f\left(\hat{\mathbf{Y}}_{2k+1}\right)$ is a $\left(W \bullet H\right)\times\left(\left(2l+1\right)\bullet\left(2l+1\right)\right)$ dimensional matrix, whose element is computed according to (2), and $\boldsymbol{\alpha}$ is the same with that of (2). Incorporating (2) into (4), we get the approximation of $\hat{\mathbf{X}}_{2k+2}$, where the weight is non-linear with respect to AR coefficient $\boldsymbol{\alpha}$, and the expression can be presented as

$$\hat{\mathbf{X}}_{2k+2} = f\left(f\left(\mathbf{X}_{2k}\right)\boldsymbol{\alpha}\right)\boldsymbol{\alpha} = g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right) + \mathbf{n}_{2k+2}, \tag{5}$$

where $g\left(\mathbf{X}_{2k}\right)$ is a function which transfers $\mathbf{X}_{2k}$ to a $\left(W \bullet H\right)\times\left(\left(4L+1\right)\bullet\left(4L+1\right)\right)$ dimensional matrix, $\mathbf{W}\left(\boldsymbol{\alpha}\right)$ is the newly weight vector corresponding to the enlarged spatial neighborhood, shown as the circles in Fig. 4, in which the weight element is the quadratic of the AR coefficient $\alpha_{i,j}$. This process can be interpreted as in Fig. 4, where each pixel in the backward reference frame $2k+2$ can be interpolated by a combination of pixels in an enlarged spatial neighborhood with size of $\left(4L+1\right)\times\left(4L+1\right)$ in the forward reference frame $2k$, shown as the circles in Fig. 4, after the propagation of AR interpolation in temporal axis along the motion trajectory. Consequently, the length of the weight vector $\mathbf{W}\left(\boldsymbol{\alpha}\right)$ of the enlarged spatial neighborhood is $\left(4L+1\right)\times\left(4L+1\right)$, and $g\left(\mathbf{X}_{2k}\right)$ is a $\left(W \bullet H\right)\times\left(\left(4L+1\right)\bullet\left(4L+1\right)\right)$

matrix, where each row vector $g_m\left(\mathbf{X}_{2k}\right)$, $m = 0,1,\ldots,W \times H - 1$, corresponds to the enlarged AR spatial neighborhood of the pixel m.

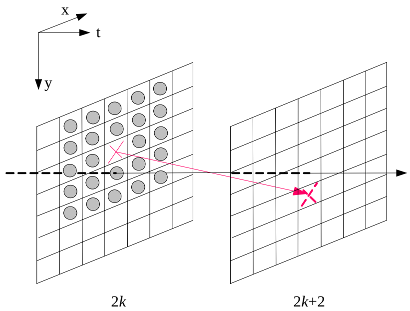

Fig. 4 the AR interpolation between the forward and backward reference frames along the motion trajectory after the pro pagation of the AR interpolation in the intermediate frame

Given the image model in Eq. (5), the coefficient vector $\boldsymbol{\alpha}$ can be obtained by minimizing the following mean squared error

$$\begin{aligned}\varepsilon\left(\boldsymbol{\alpha}\right) &= \frac{1}{2}\left(\mathbf{X}_{2k+2} - g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right)\right)^t\left(\mathbf{X}_{2k+2} - g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right)\right) \\ &= \frac{1}{2}\left\|\vec{r}\left(\boldsymbol{\alpha}\right)\right\|^2\end{aligned}, \quad (6)$$

where $\vec{r}\left(\boldsymbol{\alpha}\right) = \mathbf{X}_{2k+2} - g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right)$ is defined to be the residual vector. The basic idea behind this method is that the original image should satisfy the equation $\partial\varepsilon/\partial\boldsymbol{\alpha} = 0$. However, $\varepsilon\left(\boldsymbol{\alpha}\right)$ is not linear with respect to $\boldsymbol{\alpha}$. To solve this problem, we derive a linear form for $\vec{r}\left(\boldsymbol{\alpha}\right)$. Let $\Delta\boldsymbol{\alpha}$ represent a small change in the AR coefficient vector $\boldsymbol{\alpha}$, and then express $\vec{r}\left(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\right)$ in Tayer series as

$$\begin{aligned}&\vec{r}\left(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\right) \\ &= \mathbf{X}_{2k+2} - g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\right) \\ &\approx \mathbf{X}_{2k+2} - g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right) - J\left(\boldsymbol{\alpha}\right)\Delta\boldsymbol{\alpha} \\ &= \vec{r}\left(\boldsymbol{\alpha}\right) - J\left(\boldsymbol{\alpha}\right)\Delta\boldsymbol{\alpha}\end{aligned}, \quad (7)$$

where $J\left(\boldsymbol{\alpha}\right)$ is the Jacobian of $g\left(\mathbf{X}_{2k}\right)\mathbf{W}\left(\boldsymbol{\alpha}\right)$ with respect to $\boldsymbol{\alpha}$. Therefore the minimization problem in (6) can be modified as

$$\begin{aligned}\varepsilon\left(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\right) &= \frac{1}{2}\left\|\vec{r}\left(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\right)\right\|^2 \\ &= \min_{\Delta\boldsymbol{\alpha}}\left\|J\left(\boldsymbol{\alpha}\right)\Delta\boldsymbol{\alpha} - \vec{r}\left(\boldsymbol{\alpha}\right)\right\|^2\end{aligned}. \quad (8)$$

By using this method, the original problem for direct estimation of $\boldsymbol{\alpha}$ has been transformed into the minimization problem for the increment $\Delta\boldsymbol{\alpha}$ in (8). The proposed iterative algorithm to estimate the optimum vector $\boldsymbol{\alpha}$ is summarized in Table 1.

Table 1. Summary of the proposed iterative algorithm

Step 1. Initialize $\boldsymbol{\alpha}^0$.

Step 2. Compute $\vec{r}\left(\boldsymbol{\alpha}^0\right)$ and $J\left(\boldsymbol{\alpha}^0\right)$.

Step 3. At the *i*-th iteration, compute $\Delta\boldsymbol{\alpha}^i$ according to the following equation.

$$\left[J^t\left(\boldsymbol{\alpha}^i\right)J\left(\boldsymbol{\alpha}^i\right)\right]\times\left(\Delta\boldsymbol{\alpha}^i\right)=J^t\left(\boldsymbol{\alpha}^i\right)\vec{r}\left(\boldsymbol{\alpha}^i\right) \tag{9}$$

Step 4. Update the estimate of $\boldsymbol{\alpha}^{i+1}$ with

$$\boldsymbol{\alpha}^{i+1}=\boldsymbol{\alpha}^i+\Delta\boldsymbol{\alpha}^i \tag{10}$$

Step 5. Update $\vec{r}\left(\boldsymbol{\alpha}^{i+1}\right)$ and $J\left(\boldsymbol{\alpha}^{i+1}\right)$.

Step 6. Go to Step 3 and update $i=i+1$ until convergence or a maximum number of iterations is reached.

The similar iterative algorithm can also be applied to derive the backward AR coefficient $\boldsymbol{\beta}$. After that, the final predicted pixel value within frame $2k+1$ can be computed as

$$\hat{\mathbf{Y}}_{2k+1}=\left(f\left(\mathbf{X}_{2k}\right)\boldsymbol{\alpha}+f\left(\mathbf{X}_{2k+2}\right)\boldsymbol{\beta}\right)/2. \tag{11}$$

4. EXPERIMENTAL RESULTS

To demonstrate the superiority of the proposed 3D AR model for the bi-directional prediction, the 3D AR model is utilized to replace the traditional direct mode in video coding. That is to say, for each Macroblock (MB) in B picture, the traditional direct mode is disabled but in contrary, the 3D AR model is implemented to replace it, and the 3D AR coefficients are derived by the proposed iterative method. Since the derivation of the 3D AR coefficients is performed in the forward and backward reference frames, there is no mismatch in the decoder, and consequently there is no need to transmit the AR coefficients. The initialization of $\boldsymbol{\alpha}$ is as follows. We first set the initial pixel value $\hat{Y}_{2k+1}^0(m,n)$ as the values derived by the direct mode. Then initial $\boldsymbol{\alpha}^0$ can then be computed according to

$$\boldsymbol{\alpha}^0=\min_{\boldsymbol{\alpha}}\left(\left(f\left(\mathbf{X}_{2k}\right)\boldsymbol{\alpha}-\hat{\mathbf{Y}}_{2k+1}^0\right)^t\left(f\left(\mathbf{X}_{2k}\right)\boldsymbol{\alpha}-\hat{\mathbf{Y}}_{2k+1}^0\right)+\left(f\left(\hat{\mathbf{Y}}_{2k+1}^0\right)\boldsymbol{\alpha}-\mathbf{X}_{2k+2}\right)^t\left(f\left(\hat{\mathbf{Y}}_{2k+1}^0\right)\boldsymbol{\alpha}-\mathbf{X}_{2k+2}\right)\right). \tag{12}$$

The maximum iteration number is set to be 5, and the criterion for converging is to judge whether the distortion of the interpolated blocks between two successive iterations is smaller than a threshold. In the experiment, the threshold is set to be 50.

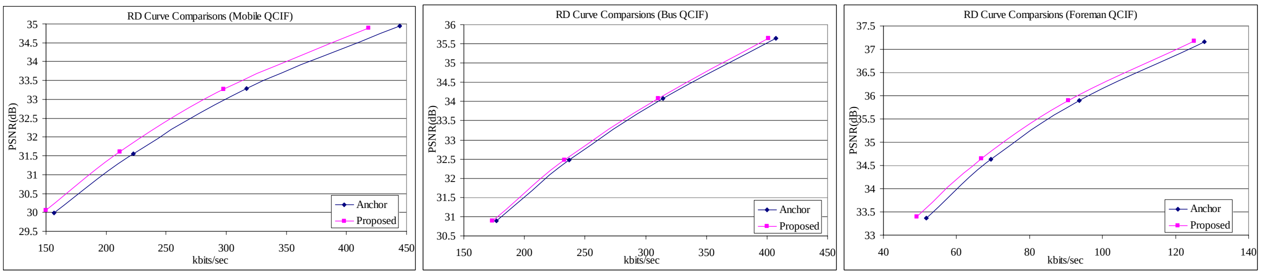

Fig. 5 Rate-distortion curves for sequences *Mobile*, *Bus* and *Foreman*

The proposed method is implemented based on H.264/AVC reference software JM98. The test sequences include *Foreman*, *Mobile*, and *Bus* in the QCIF format with 30fps, and 200 frames of each test sequence are encoded. The spatial direct mode (SDM) is chosen as the anchor to compare the performance by the proposed 3D AR model and the encoding structure is IBPBP…… To evaluate the average PSNR vs bit-rate, the method described in [5], which is widely used during H.264/AVC development, is employed.

The overall RD performances (include I, P, B pictures) are shown in Fig. 5. It's easy to observe that the coding performance is improved by the proposed 3D AR model, compared with the SDM, from low to high bit rates for all the test sequences. The ratio of the direct mode, by the proposed method and the anchor is presented in Table 2. Here if one MB is finally predicted by the 3D AR model after the RDO selection, it is also considered to be direct mode since there is no need to transmit motion vector difference in 3D AR model. From Table 2, it can be seen that the ratio of the MBs encoded with the direct mode by the proposed 3D AR model is always higher than that of the SDM except for the *Bus* sequence. In other words, it further verifies the proposed 3D AR model is able to bi-directionally predict the frame with higher accuracy. That's because when applying the proposed 3D AR model, higher accuracy prediction is able to be achieved while the overhead does not increase, and thus more MBs will be encoded by the 3D AR model when RDO is enabled. Table 3 presents the detailed experimental results on all the test sequences. It reveals that the performance of the proposed 3D AR model outperforms the SDM in H.264. The maximum PSNR gain can be up to 1.227 dB in terms of B frames and 0.28 dB for total frames. The maximum bit rate saving is up to 39.942% in terms of B frames and 5.696% for the total frames. The excellent performance is largely attributed to the high accuracy of the proposed 3D AR model for bi-directional prediction. Furthermore, since the RDO is enabled, many MBs which do not select the direct mode as the optimum one when applying the traditional SDM now prefers the direct mode because of the better prediction performance provided by the proposed 3D AR model. Consequently, it further reduces the bits which would be used to describe the mode, motion vector as well as residual information if other mode is selected.

Table 2 Ratio of MBs with direct mode

QP	*Foreman*		*Mobile*		*Bus*	
	Anchor	AR	Anchor	AR	Anchor	AR
26	50.258%	58.132	60.997%	73.367%	31.375%	32.654%
28	57.349%	68.295%	64.770%	81.138%	36.383%	34.694%
30	63.410%	78.108%	67.697%	86.806%	41.662%	38.364%
32	69.058%	85.282%	67.760%	87.611%	45.496%	42.177%

Table 3 Performance comparison between the proposed 3D AR model and SDM

Video Sequences		*Foreman*	*Mobile*	*Bus*
All pictures	Average PSNR Gain	0.156	0.28	0.07
	Average Bits Saving	3.772%	5.696%	1.383%
B pictures	Average PSNR Gain	0.756	1.227	0.226
	Average Bits Saving	22.542%	39.942%	4.836%

5. CONCLUSION

This paper proposes a 3D AR model for bi-directional prediction in video coding. Applying the 3D AR model, each pixel is predicted as a weighted summation of corresponding pixels along the motion trajectories within the reference frame. The ultimate prediction of one pixel is obtained by the combination of pixels generated by the 3D AR models in both the forward and backward reference frames. To derive more accurate AR coefficients, an iterative algorithm using a nonlinear least squares method is proposed. In contrary to other methods that treat the AR coefficients estimation and AR interpolation as disjoint processes, the proposed iterative method enables the AR coefficients estimation and AR interpolation to be performed simultaneously. Experimental results show that the proposed method has a superior performance in video coding.

6. ACKNOWLEDGEMENTS

This work was supported in part by National Science Foundation (60833013) and Major State Basic Research Development Program of China (973 Program, 2009CB320903).

REFERENCES

[1] S.W.Wu and A. Gersho, "Joint estimation of forward and backward motion vectors for interpolative prediction of video," EEE Trans. Image process., vol.3, no.5, pp: 684-687, Sept.1994.

[2] Video Codec for Audiovisual Services at p×64 kbit/s, 1990. ITU-T Rec. H.261.

[3] Video Coding for Narrow Telecommunications Channels at <64 kbit/s, 1996. ITU-T Rec. H.263.

[4] B. Girod, E. Steinbach, and N. Farber, "Performance of the H.263 video compression standard," J. VLSI Signal Process., no. 17, pp. 101-111, 1997.

[5] T. Wiegand, GJ Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol, vol. 13, No. 7, pp: 560-576, July. 2003.

[6] G.J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," Proc. IEEE, vol.93, no. 1, pp. 18-31, 2005.

[7] M. Flierl, T. Wiegand, and B. Girod, “ A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction,” in Proc. IEEE DCC, 1998, pp. 239 -248.

[8] B. Girod, “Efficiency analysis of multihypothesis motion-compensatedprediction for video coding,” IEEE Trans. Image process., vol.9, no.2, pp: 173-183, Feb.2000.

[9] X. Wu, K. U. Barthel, and W. Zhang, “Piecewise 2D autoregression for predictive image coding,” in Proc. IEEE Int. Conf. Image Processing, ICIP, 1998, pp. 901 -904.

[10] Y. Zheng and X. Li, "Video modeling via spatio-temporal adaptive localized learning (STALL)," Signals, Systems and Computers, 2006. ACSSC '06, pp. 779-783.

[11] X. Li, "Least-square prediction for backward adaptive video coding," EURASIP Journal on Applied Signal Processing, vol. 2006, pp. 1-13, 2006.