LOCAL ADAPTIVE LEARNING AND FUSION FOR SIDE INFORMATION INTERPOLATION IN DISTRIBUTED VIDEO CODING

Xianming Liu[1]*, Yongbing Zhang*[1]*, Yongpeng Li*[2]*, Hongbin Liu*[1]*, Siwei Ma*[3]*, Debin Zhao*[1]
[1]Department of Computer Science and Technology, Harbin Institute of Technology, Harbin,150001, P.R. China
[2]Graduate University, Chinese Academy of Science, Beijing, 100039, P.R. China
[3]Institute of Digital Media, Peking University, Beijing, 100871, P.R. China
xmliu@jdl.ac.cn

ABSTRACT

Motivated by theoretical analysis of the curve fitting problem based on equivalent kernel, in this paper we propose a local adaptive learning and fusion model for side information interpolation in distributed video coding. In the proposed model, each pixel in the interpolated frame is approximated as the linear combination of samples within a local spatio-temporal window using kernel parameters as weight. The size of training window can be adaptive to the motion characteristic of video, from samples in which the kernel parameters can be locally learned. In order to further improve the quality of interpolated frames, we introduce a belief-projection based fusion strategy with adaptive weights for multiple interpolated results which are with the same time index. Experimental results demonstrate that the proposed learning and fusion model is effective in performance for side information interpolation in distributed video coding.

Index Terms— Side information interpolation, distributed video coding, equivalent kernel, spatio-temporal local adaptive learning, fusion

1. INTRODUCTION

Traditional video compression standards, such as H.264 and MPEG-4, are based on inter-frame predictive coding in order to exploit temporal correlation between successive frames. Given that predictive coding uses motion estimation, which is typically a high complex process, the video encoder is typically more complex than the decoder. However, due to limited energy and computing ability, several emerging applications like wireless video surveillance and mobile camera phones can not afford such a high complex encoder. As a consequence, a low complexity encoder with high coding efficiency is much desirable. Distributed video coding (DVC) is a new video coding paradigm which can shift the complexity from encoder to decoder by intra-frame encoding and inter-frame decoding. It has been proved in theory [1],[2] that distributed source coding can achieve the same coding efficiency as jointly encoding. So the field of DVC research has been receiving more and more attention in recent years.

Side information (SI) quality is an important factor determining the coding efficiency of the DVC system. One of the most popular methods for generating SI is motion compensated temporal interpolation (MCTI), in which motion information is explicitly estimated from adjacent frames at the decoder by block-matching-based motion searching. The apparent advantage of MCTI is its conceptual simplicity, and block-matching can reflect some relationship between motion and interpolated intensity values, especially when the motion accords with the translation model. However, due to the original frames are not available at the decoder, block matching may not be effective locally, which usually results in artifacts in SI. Although some complex post-process steps, e.g. overlapped block motion compensation (OBMC) or spatial motion vector smoothing [3], have been used to improve the quality of SI, it is still far from satisfactory. As a consequence, some locally accurate motion models are popular in the process of interpolation. An alternative perspective for motion estimation is that motion information can be wisely derived in a filter-based fashion [4],[5]. Such localized estimation can be viewed as an implicit approach of exploiting motion-related temporal dependency, in which motion information is embedded into predictive coefficients trained.

In this paper, we propose an alternative approach for side information generation based on a local adaptive learning and fusion model. We formulate the frame interpolation as a curve fitting problem, namely, fitting the intensity curve of side information according to the coordinates and intensity values of adjacent reconstructed frames. The fitting kernel parameters are adaptively learned within a localized spatio-temporal window on a pixel-by-pixel basis. In order to further improve the quality of interpolated frames, we introduce an adaptive weight fusion method for multiple interpolated results with the same time index.

The rest of this paper is organized as follow. In Section 2, we give a theoretical analysis of the curve fitting problem, which can bring us some motivation for frame interpolation. We describe our proposed model in detail in Section 3. In Section 4, the experimental results are presented to show the efficiency of our model and Section 5 concludes this paper.

2. CURVE FITTING BASED ON EQUIVALENT KERNEL

The frame interpolation can be formulated as a curve fitting problem. Les us now consider the curve fitting problem based on equivalent kernel, which can motivate a number of key concepts for the presented local adaptive learning and fusion based side information interpolation in this paper [6].

The goal of curve fitting is to exploit the training set to predict the value t of the target variable for a new value x of the input variable. Consider a model defined in terms of a linear combination of *M* basis functions given by elements of the vector $\phi(x)$ so that

$$y(x,w)=w^T\phi(x)\,, \tag{1}$$

where x is the input vector and w is *M*-dimensional weight vector. We assume that the target variable t is given by a deterministic function $y(x,w)$ with additive Gaussian noise so that

$$t=y(x,w)+\varepsilon\,, \tag{2}$$

where ε is a zero mean Gaussian random variable with precision (inverse variance) β . Consider N inputs $X=(x_1,...,x_n)^T$ with corresponding target values $T=(t_1,...,t_n)^T$ .For (2), the likelihood function is:

$$p(\mathrm{T}\,|\,X,w,\beta)=\prod_{n=1}^{N}\mathrm{N}(t_n\,|\,w^T\phi(x_n),\beta^{-1})\,, \tag{3}$$

and the Gaussian prior is :

$$p(w)=\mathrm{N}(w\,|\,0,\alpha^{-1}I) \quad , \tag{4}$$

with covariance $\alpha^{-1}I$, where α is the precision and I is the identity matrix. According to (3) and (4), the Bayesian treatment of linear regression model is formulated as:

$$p(w|\,X,\mathrm{T})=p(\mathrm{T}\,|\,X,w,\beta)\,p(w) \ , \tag{5}$$

From this general result we obtain the posterior distribution over w:

$$p(\mathrm{w}|\,X,\mathrm{T})=\mathrm{N}(w\,|\,m_N,S_N) \ , \tag{6}$$

where

$$m_N=\beta S_N\Phi^T T\,, \tag{7}$$

$$S_N^{-1}=\alpha I+\beta\Phi^T\Phi\,, \tag{8}$$

with $\Phi=(\phi(x_1),\cdots,\phi(x_N))^T$. Log of (6) can be expressed as a function of w:

$$\ln p(w\,|\,X,T)=-\frac{\beta}{2}\sum_{n=1}^{N}\{t_n-w^T\phi(x_n)\}^2-\frac{\alpha}{2}w^T w+const\,. \tag{9}$$

We solve this function with respect to w using maximum likelihood. However, in practice, we are not usually interested in the value of w itself but rather in making predictions of t for new values of x . The posterior mean solution (7) has an interesting interpretation that will set the stage for kernel methods. We substitute (7) into the expression (1) and get:

$$y(x,m_N)=m_N^T\phi(x)=(m_N^T\phi(x))^T=\sum_{n=1}^{N}\phi(x)^T\beta S_N\phi(x_n)t_n\,. \tag{10}$$

Thus the mean of the predictive distribution at a point x is given by a linear combination of the training set target variables t_n , so

$$y(x,m_N)=\sum_{n=1}^{N}k(x,x_n)t_n\,, \tag{11}$$

where $k(x,x_n)$ is known as the equivalent kernel defined by:

$$k(x,x_n)=\beta\phi(x)^T S_N\phi(x_n)=\psi(x)^T\psi(x_n) \ , \tag{12}$$

where $\psi(x)=\beta^{\frac{1}{2}}S_N^{\frac{1}{2}}\phi(x)$ is a vector of x .

Generalizing the analysis result to side information interpolation, we propose a local adaptive learning and fusion model, which will be described in detail in next section.

3. THE PROPOSED LOCAL ADAPTIVE LEARNING AND FUSION MODEL

3.1. Local Adaptive Learning Model

As well known, there is high similarity between successive frames in video sequences, namely that pixels neighboring in a local spatio-temporal space possess an underlying regularity, which we want to learn. In the problem of frame interpolation, the motion regularity is implicitly embedded into kernel parameters.

Suppose $\{X(x,y,t)\}$ is the given video sequence, where $(x,y)\in[1,H]\times[1,W]$ are spatial coordinates and $t\in[1,T]$ is the frame index. We denote the position of a pixel in side information by a vector $\overrightarrow{n_0}=(x,y,t)$, which represents the input variable. The intensity value $\hat{X}\left(\vec{n_0}\right)$ denotes the target variable. The training set of $\overrightarrow{n_0}$ in the proposed model not only includes the pixels within its two temporal neighborhoods taken in the previous and the following frames but also the available interpolated pixels within its spatial neighborhood taken in the current frame, which is different from the setting in [5]. Each temporal neighborhood is specified as a $(2L+1)\times(2L+1)$ region bounded by $(x\pm L,y\pm L)$, where L is the spatio-temporal order. Consequently, the spatial neighborhood is specified as a region with the size of $\lfloor(2L+1)\times(2L+1)/2\rfloor$. Thus the model order N is:

$$N=2\times(2L+1)\times(2L+1)+\lfloor(2L+1)\times(2L+1)/2\rfloor\,. \tag{13}$$

Fig.1illustrates the case when $L=1$.

The intensity value of $\overrightarrow{n_0}$ is approximated as the linear combination of samples using kernel parameters as weights within a localized spatio-temporal window. Therefore, (11) becomes

$$\widehat{X}\left(\overrightarrow{n_0}\right)=\sum_{i=1}^{N}k\left(\overrightarrow{n_0},\overrightarrow{n_i}\right)X\left(\overrightarrow{n_i}\right) \tag{14}$$

where $k(n,n')$ are kernel parameters, which should be adaptively updated within the spatio-temporal window.

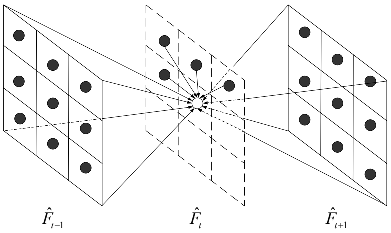

Fig.1 Local-adaptive-learning model for side information interpolation with $L=1$

The efficiency of the model described in (14) heavily relies on the choice of the size of the training window and the spatio-temporal order. Intuitively, the training window should cover the support of the probability density function (*pdf*) of motion vectors. In our experiment, the window size is empirically set to 16×16. The spatio-temporal order is closely related to the motion in the training window, because smaller spatio-temporal order will achieve good performance for stationary regions; however, for moving regions, larger spatio-temporal order is necessary. In our experiment, the motion vector in MCTI can be utilized to measure the motion in the training window. The spatio-temporal order of the model is computed by

$$L=\max_{block_i\in S}\left\{abs\left(\left\lfloor mvx_{block_i}\right\rfloor\right),abs\left(\left\lfloor mvy_{block_i}\right\rfloor\right)\right\}+1 \tag{15}$$

where S represents the training window, mvx_{block_i} and mvy_{block_i} represent the horizontal and vertical motion vectors of the *i*th 8×8 block after performing MCTI.

3.2. Adaptive Weight Fusion of Interpolated Results

An adaptive weight fusion strategy is also proposed to further improve the quality of interpolated frames. In order to avoid choosing basic functions and to derive more accurate kernel parameters, we utilize a so-called self-feedback method to do training [5]. We group five successive frames as an interpolation unit (IU) to jointly train kernel parameters, which slides along the time axis at a step of GOP size as depicted in Fig.2. As a consequence, there are multiple interpolated results with the same time index but belonging to different IU. For instance, there are two interpolated results $\widehat{X^{IUF}_{\text{t+1}}}$ and $\widehat{X^{IUB}_{\text{t+1}}}$ at time *t*+1, one is from *IUF* and another is from *IUB*. Note that $\widehat{X^{IUF}_{\text{t+1}}}$ contains more forward motion information as its kernel parameters are trained from frames $X_{\text{t}-2}$, X_t and X_{t+2}, while $\widehat{X^{IUB}_{\text{t+1}}}$ contains more backward motion information as its kernel parameters are trained from frame X_t, X_{t+2} and X_{t+4}. We consider that better interpolation performance can be achieved by adaptively fusing these two interpolated results compared with choosing an optimal one. Such an idea is similar to *B* frame coding in H.264. The only difference is that *B* frame generates the prediction frame by averaging the forward and backward reference frames while our strategy is to weight $\widehat{X^{IUF}_{\text{t+1}}}$ and $\widehat{X^{IUB}_{\text{t+1}}}$ adaptively. The fused side information can be written as:

$$S_t(\overrightarrow{n_0})=\alpha(\overrightarrow{n_0})\widehat{X^{IUF}_{\text{t+1}}}(\overrightarrow{n_0})+\left(1-\alpha(\overrightarrow{n_0})\right)\widehat{X^{IUB}_{\text{t+1}}}(\overrightarrow{n_0})\,, \tag{16}$$

where α is the fusion weight. It is intuitive that the efficiency of fusion mainly depends on the choice of α. We utilize a belief-projection strategy to determine α. Note that we re-interpolated frame *t* in *IUF* in the process of training kernel parameters [5]. That can be formulated as follow:

$$\widehat{X^{IUF}_{t}}=Learning(\widehat{X^{IUF}_{t-1}},\widehat{X^{IUF}_{t+1}},k^{IUF})\,. \tag{17}$$

where the function $Learning(\bullet)$ denotes the learning process presented in subsection 3.1. The interpolated frame *t*-1 and trained kernel parameters in *IUF* along with interpolated frame *t*+1 in *IUB* are also used to re-interpolate frame *t,* as illustrated in Fig.2.

$$\widehat{X^{IUB}_{t}}=Learning(\widehat{X^{IUF}_{t-1}},\widehat{X^{IUB}_{t+1}},k^{IUF}) \tag{18}$$

From (17) and (18), we can see the interpolated quality of $\widehat{X^{IUF}_{t+1}}$ and $\widehat{X^{IUB}_{t+1}}$ can be directly reflected by $\widehat{X^{IUF}_{t}}$ and $\widehat{X^{IUB}_{t}}$, which is called belief-projection. First, we compute SAD value for each pixel in $\widehat{X^{IUF}_{t}}$ and $\widehat{X^{IUB}_{t}}$ using corresponding key frames as original ones. Then the SAD value of each pixel in $\widehat{X^{IUF}_{\text{t+1}}}$ and $\widehat{X^{IUB}_{\text{t+1}}}$ is determined as follow:

$$\begin{aligned}SAD_{IUF}(\overrightarrow{n_0})&=\sum_{\overrightarrow{n_0}\in neighbour(\overrightarrow{n_i})}|\,\widehat{X^{IUF}_{t}(\overrightarrow{n_i})}-X_t(\overrightarrow{n_i})\,|\\SAD_{IUB}(\overrightarrow{n_0})&=\sum_{\overrightarrow{n_0}\in neighbour(\overrightarrow{n_i})}|\,\widehat{X^{IUB}_{t}(\overrightarrow{n_i})}-X_t(\overrightarrow{n_i})\,|\end{aligned}\;, \tag{19}$$

where $neighbour(\overrightarrow{n_i})$ denotes the training set of $\overrightarrow{n_i}$ .So

$$\alpha=\frac{SAD_{IUB}}{SAD_{IUF}+SAD_{IUB}} \tag{20}$$

which is based on the fact that smaller interpolated error leads to larger weight in fusion.

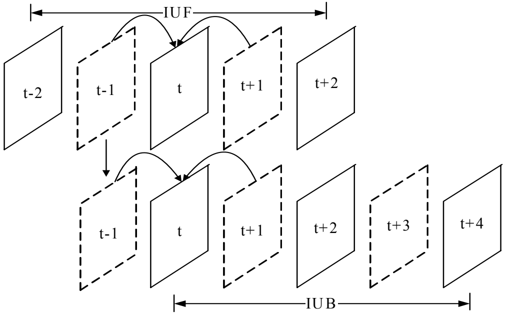

Fig.2 Adaptive fusion strategy for multi interpolated results

Table 1 Objective quality comparison for interpolated side information(in dB)

	Foreman			Mobile		
	QP=28	QP=24	QP=20	QP=28	QP=24	QP=20
MCTI_OBMC	34.49	35.56	36.44	32.80	34.22	34.76
LearningWithoutFusion	34.59	35.82	36.78	33.21	35.10	35.84
ComparedBMA	34.90	36.21	36.92	33.40	35.22	35.98
LearningWithFusion	**35.31**	**36.65**	**37.66**	**33.49**	**35.40**	**36.19**
Gain(overMCTI_OBMC)	**0.82**	**1.09**	**1.22**	**0.69**	**1.18**	**1.43**

4. EXPERIMENTAL RESULTS

In this section, experimental results are provided to demonstrate the performance of the proposed model. Results of two test sequences including *Foreman* and *Mobile* (QCIF, 30Hz, 4:2:0) are presented. In each sequence, 100 frames are selected and the GOP structure is IWI, where key frames are encoded with H.264 intra coding method and WZ frames are encoded with the pixel-domain turbo code based WZ codec [7].

Table 1 includes the objective performance comparison for interpolated side information with three QP values: 20, 24, 28, where

- MCTI_OBMC: the block-matching-based motion compensation temporal interpolation, OBMC is used as a post process to smooth motion field.
- LearningWithOutFusion: local adaptive learning without fusion.
- ComparedBMA: the result of MCTI_OBMC is utilized as a criterion to determine which is better for two regression results.
- LearningWithFusion: the method we proposed in this paper.

From Table 1 we can see that our approach significantly outperforms the MCTI_OBMC approach. Our method can improve up to 1.43dB for *Mobile* sequence and 1.22dB for *Foreman* sequence, respectively.

We further test the efficiency of our model in terms of overall performance in DVC scheme. Simulation results presented in Fig.3 show that our model can improve 0.6dB for *Foreman* sequence at most and 1dB for *Mobile* sequence, respectively.

5. CONCLUSION

In this paper, we have proposed a novel local adaptive learning and fusion model for side information interpolation. According to the duality of curve fitting and frame interpolation, we first analyze the cure fitting problem based on equivalent kernel in theory. The analysis result is then generalized to our proposed learning model. In order to further improve the quality of interpolated frame, we introduce a belief-projection based adaptive weight fusion strategy. Experimental results demonstrate that our learning and fusion model can significantly improve the coding efficiency of the DVC system.

6. ACKNOWLEDGEMENTS

This work was supported in part by National Science Foundation (60672088 and 60736043) and National Basic Research Program of China (973 Program, 2009CB320905).

7. REFERENCES

[1] D.Slepian and J. K. Wolf, “Noiseless coding of correlated information sources”, IEEE Transactions on Information Theory, pp.471-480, Vol. 19, July.1973.

[2] A.Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder, “ IEEE Transactions on Information Theory, Vol. 22, pp.1-10, Jan. 1976.

[3] J. Ascenso, C. Brites, and F. Pereira, “Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding”, EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services, Slovak, July 2005.

[4] M.Szummer and R.W.Picard, “Temporal texture modeling,” in Proc. IEEE Int. Conf. Image Processing, ICIP, 1996, pp. 823–826.

[5] Y. Zhang, D. Zhao, X. Ji, R. Wang, and X. Chen, “A spatial-temporal autoregressive frame rate up conversion,” in Proc. IEEE Int. Conf. Image Processing, ICIP, 2007, pp. 441 -444.

[6] M.Bishop, Pattern recognition and machine learning. New York: Springer. (2006).

[7] B. Girod, A. Aaron, S. Rane, and D. R. Monedero, “Distributed video coding,” *Proc.* IEEE, vol. 93, no.1, pp.71–83, Jan.2005

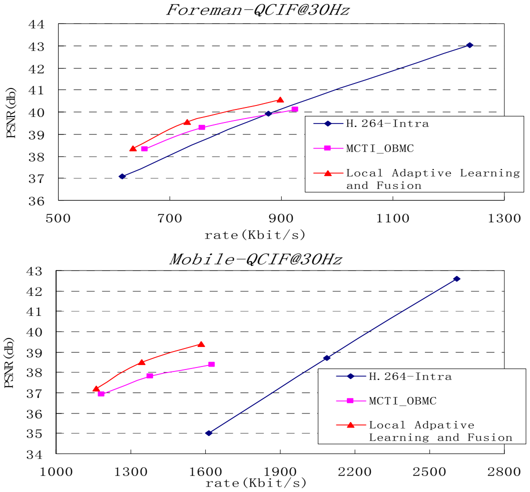

Fig. 3 Simulation results for *Foreman* and *Mobile*