SIDE INFORMATION EXTRAPOLATION WITH TEMPORAL AND SPATIAL CONSISTENCY

Xianming Liu[1], *Deming Zhai*[1],*Debin Zhao*[1], *Ruiqin Xiong*[2], *Siwei Ma*[2], *Wen Gao*[2]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin,150001, P.R. China
[2]School of Electronic Engineer and Computer Science, Peking University, Beijing,100871, P.R. China

ABSTRACT

In this paper, we present an efficient side information extrapolation scheme with temporal and spatial consistency for low-delay Wyner-Ziv video coding. Our method is based on the regularized local linear regression (RLLR) model, in which each pixel in SI is approximated as a linear weighted combination of samples within a local temporal neighborhood. The optimal model parameters are estimated by projecting the transformation function onto the temporal training samples to exploit motion-related dependency. During this procedure, moving weights are incorporated into the objective function to express the relative importance of training samples in estimating parameters of the model. Furthermore, spatial correlation is explored by imposing an additional local smoothness penalty, which does good to estimate the occluded regions and complex motion regions. The learned function is smooth and locally linear, and can be obtained with a closed-form solution by solving a convex optimization problem. Experimental results demonstrate that the RLLR method achieves very competitive SI extrapolation performance compared with the state-of-the-art methods.

***Index Terms*—** Distributed video coding, side information extrapolation, temporal and spatial consistency, regularized local linear regression

1. INTRODUCTION

Emerging as an enabling technology for wireless sensor networks, distributed source coding (DSC) has received more and more attention in recent years. It refers to compress correlated signal source captured by different sensors which do not communicate between themselves. All signals captured are encoded independently and transmitted to a central base station, where they are decoded jointly. Slepian-Wolf [1] and Wyner-Ziv [2] had proved that under certain conditions separate encoding does not induce any compression efficiency loss when compared to the joint encoding used in the traditional predictive coding paradigm. Video coding has been recast into the distributed source coding framework, leading to distributed video coding (DVC) [3]. Compared to the state-of-the-art video coding standard, such as H.264 and MPEG-4, DVC has a lower computation complexity and error resilience. Such property is conceptually appealing for some practical video applications, such as wireless video surveillance and mobile camera phones.

One of crucial factors to influence the performance of DVC system is the quality of side information (SI). In [4], two typical SI generation approaches are suggested, which make use of interpolation and extrapolation. For interpolation, SI for the current frame is obtained by using the adjacent previously and subsequently decoded frames. However, in low-delay application, the temporally subsequent frames cannot be used as references to generate SI. Therefore, extrapolation is suitable for low-delay applications.

In the literature, many approaches have been proposed to improve the performance of SI extrapolation. In [5], a robust extrapolation module is proposed to generate SI based on motion field smoothing. S. Borchert *et al.* [6] introduce a true motion based extrapolation scheme considering the 3-D recursive search (3DRS) motion estimation. These methods are all based on a translational motion model, in which it is assumed that the motion in the current frame is a continuous extension of the motion in the previous frame. However, the translation model is not always satisfied, especially for the video sequences with high motion. Alternatively, Zhang *et al.* [7] generalize the corresponding problem in explicit ME into an adaptive filtering problem, and achieve promising results. In this scheme, motion information is no longer represented explicitly as motion vectors but implicitly embedded into the filter coefficients.

In this paper, we propose a more efficient SI extrapolation method based on regularized local linear regression (RLLR). Our algorithm simultaneously minimizes the moving least squares error on the temporal training samples and preserves the local spatial geometrical structure of the same frame, therefore achieves wonderful temporal and spatial consistency. By fully exploiting the spatio-temporal correlation information among the neighboring space, the obtained transformation functions are smooth and locally linear, and can keep the local motion information better. Compared to the work presented in [7], the contribution of our work is highlighted as follows: (1) the estimation granularity is pixel level, at which motion information can be represented more accurately; (2) moving weights are introduced in the objective function, which can efficiently handle the influence of statistical outliers and lead to a robust estimator; (3) spatial

978-1-4244-9472-9/11/$26.00 ©2011 IEEE

correlation is exploited by incorporating an additional penalty term, which does good to estimate the occluded regions and complex motion regions.

The rest of this paper is organized as follows. Section 2 presents the framework of regularized local linear regression. Section 3 discusses the algorithm details and gives some discussion about the proposed model. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. THE FRAMEWORK OF REGULARIZED LOCAL LINEAR REGRESSION

2.1. Extrapolation Model

Extrapolation is to infer the SI according to previous reconstructed frames. Supposing $x_i \in \Re^2$ is the pixel to estimate in SI for the WZ frame F_t, the problem is how to determine its intensity y_i with minimum uncertainty from the local covering called training window $\mathcal{N}_{t-1}(x_i)$ in the previous reconstructed frame F_{t-1}. Based on geometric constraint of motion trajectory, estimation along motion orientation is optimal in the sense of best resolving the uncertainty of y_i. However, motion orientation is hard to precisely describe since it can take any real number. Alternatively, we utilize the strategy of linear weighting of pixels in the temporal neighborhood $\phi(x_i) \in \Re^{d\times 1}$, which is a subset of X_{t-1} including the N nearest temporal neighbors of x_i from all surrounding directions. Specially, we consider a linear transformation function $f_i(\cdot;\mathbf{a}_i, b_i)$ defined as follows:

$$f_i(x_i) = \mathbf{a}_i^T \phi(x_i) + b_i \qquad (1)$$

where $\mathbf{a}_i$ is a transformation vector; b_i is a translation constant; $f_i(x_i)$ is the estimated intensity value of x_i.

The extrapolation model is illustrated in Fig. 1, where the temporal neighborhood is 3 × 3 and the training window is 7 × 7. The hollow circle represents the pixel to estimate. The yellow ones are its temporal neighbors centered on the MV aligned pixel in the forward reference frame. The yellow and gray circles construct the training window together.

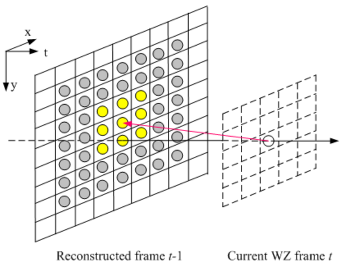

Fig. 1. Extrapolation model.

2.2. Regularized Local Linear Regression

Our goal is to estimate the optimal transformation function f_i using temporal-spatial correlation information contained in the neighboring space around the pixel x_i. Since the function f_i indexed by i is defined for each pixel point but not shared by all samples globally, we refer to it as *local linear regression*. As a consequence, the estimation granularity in our method is pixel. This is different to that of the work [7], where all pixels in one block share the same model parameters therefore the granularity level is block.

Let us try to view the SI extrapolation problem under the framework of local linear regression. Suppose we are given a training window $\mathcal{N}_{t-1}^{x_i} = \{x_1, \ldots, x_l\}$ in the previous reconstructed frame F_{t-1} with intensity values $Y_{t-1} = \{y_1, \ldots, y_l\}$. The optimal affine transformation function f_i is found by projecting itself onto samples in training window and minimizing the following loss function:

$$J(f_i) = \sum_{x_j \in \mathcal{N}_{t-1}^{x_i}} \theta_{ij} ||y_j - f_i(x_j)||^2 + \lambda ||f_i||^2, \qquad (2)$$

where the regression term is the well-known moving least squares [8]; the second term is the shrinkage constraint, also known as the Tikhonov regularizer, which helps to improve the generalization of the solutions. θ_{ij} is the moving weight reflecting the similarity between x_i and x_j. Moving weights are incorporated in the loss function in order to express the relative importance of the image samples in estimation of model parameters. In the work presented in [7], all points in the temporal local neighborhood are considered equally important, therefore motion information may be smoothed and does not appear as sharp as they should.

The induced loss function defined above only expresses temporal relationships between two successive frames. It is more reasonable to further exploit the spatial correlation between the pixel to interpolate and previously reconstructed pixels in the current frame. Considering the smoothness of the mapping functions, an additional regularization term is imposed onto the objective function. Ultimately, we formulate the objective function which exploits both temporal and spatial dependency as follows:

$$\begin{aligned} J(f_i) = & \sum_{x_j \in \mathcal{N}_{t-1}^{x_i}} \theta_{ij} ||y_j - f_i(x_j)||^2 + \lambda ||f_i||^2 \\ & + \eta \sum_{x_p \in \mathcal{N}_t^{x_i}} s_{ip} ||f_i(x_i) - f_p(x_p)||^2 \end{aligned} \qquad (3)$$

where $\mathcal{N}_t^{x_i} = \{x_1, \cdots, x_{i-1}\}$ with intensity values $Y_t = \{f_1(x_1), \cdots, f_{i-1}(x_{i-1})\}$ is the set of reconstructed spatial examples in the current frame, s_{ip} is the weight reflects the similarity between x_i and x_p. The parameters $\lambda > 0$ and $\eta > 0$ control the relative contribution of two regularization terms in the objective function. As a result, the task of transformation function learning is to minimize the above cost function:

$$f_i^* = \arg\min J(f_i) \qquad (4)$$

2.3. Optimizing the Objective Functions

In the practical experiments, for the additional regularization penalty, we do not exploit all reconstructed examples in the current frame but only a subset $C_t = \{x_{i-[l/2]}, \cdots, x_{i-1}\}$ before x_i, where l is the pixel number in the training window. And we deal with the bias term b_i by appending each instance with an additional dimension

$$\Phi(x_i)^T \leftarrow [\phi(x_i)^T, 1], \mathbf{a}_i^T \leftarrow [\mathbf{a}_i^T, b_i], \quad (5)$$

then the loss function J can be rewritten as

$$\begin{aligned} J(\mathbf{a}_i) = & \sum_{x_j \in \mathcal{N}_{t-1}^{x_i}} \theta_{ij} ||y_j - \mathbf{a}_i^T \Phi(x_j)||^2 + \lambda \mathbf{a}_i^T \mathbf{a}_i \\ & + \eta \sum_{x_p \in \mathcal{N}_t^{x_i}} s_{ip} ||\mathbf{a}_i^T \Phi(x_i) - \widehat{y_p}||^2. \end{aligned} \quad (6)$$

In order to derive the optimal transformation vector $\mathbf{a}_i$, we take the derivative of the loss function J in Eq.(6) with respect to $\mathbf{a}_i$, and set the derivative to 0, the optimal $\mathbf{a}_i$ can be represented by

$$\mathbf{a}_i^T = \left(\sum_{x_j \in \mathcal{N}_{t-1}^{x_i}} \theta_{ij} y_j \Phi(x_j)^T + \eta \sum_{x_p \in \mathcal{N}_t^{x_i}} s_{ip} \widehat{y_p} \Phi(x_i)^T \right)$$
$$\left(\sum_{x_j \in \mathcal{N}_{t-1}^{x_i}} \theta_{ij} \Phi(x_j) \Phi(x_j)^T + \lambda \mathbf{I} + \sum_{x_p \in \mathcal{N}_t^{x_i}} s_{ip} \Phi(x_i) \Phi(x_i)^T \right)^{-1} \quad (7)$$

where $\mathbf{I}$ is the identity matrix.

The main computation burden of the above optimization process is on the inversion of a matrix in $\Re^{(d+1) \times (d+1)}$. For simplicity of representation, we denote $d = d + 1$. Let $T(d)$ be the complexity of computing the inverse of a matrix $\Re^{d \times d}$, we can get $T(d) = O(d^3)$ using standard method and $T(d) = O(d^{2.376})$ with the method of Coppersmith and Winogard. Note that in our method we set $\phi(x)$ as the 8 nearest neighboring samples of x, thus $d = 9$ (with an additional dimension for appending). As a consequence, the overall computation complexity of our method is $O(T(d)) \times N$, where N is the number of samples estimated by RLLR model. In practical experiments, we manage the computational complexity by reducing the number of samples to estimate. One way is a hybrid approach: the proposed RLLR method is only applied to pixels in blocks with occlusion or high complex motion; for pixels in blocks with smooth motion we exploit the traditional motion-compensated extrapolation algorithm [4].

3. ALGORITHM DETAILS AND DISCUSSION

In the following, let us consider a key issue in our model: the design of moving weights θ and S. And we give some discussion about the proposed RLLR model.

3.1. The Design of Moving Weights

In one frame, the image local structure is represented as a set of spatial neighboring pixels at different intensity levels. In order to preserve the edge structure well in SI extrapolation, we should consider the local structure similarity in the moving weights design. In this paper, we model a pixel and its nearest spatial neighbors in a similarity window as a vector and perform comparison on the vector instead of the single pixel. This is motivated by the idea of *non-local-means* in image processing community [9]. Note that for the current pixel to estimate its some neighbors may not be available. In practical experiments, we use the local structure of the corresponding MV aligned pixel in frame F_{t-1} as that of the current pixel.

Assuming the similarity window is sized of $K \times K$, there are totally $M = (L - K + 1)^2$ blocks in the $L \times L$ training window. We denote by $\overrightarrow{y_i}$ the column vector containing the intensity values of pixels in central $K \times K$ blocks and denote by $\overrightarrow{y_j}, j = 1, \ldots, M - 1$, the intensity vectors corresponding to the other blocks. The similarity error can be easily calculated as

$$d_{ij} = \frac{1}{m} \sum_{k=1}^{m} (\overrightarrow{y_i}(k) - \overrightarrow{y_j}(k))^2 G(k) \quad (8)$$

where $m = K \times K$ is the pixel number in the similarity window, $G(k)$ is the kernel function controlling the contribution of each pixel, which is defined as $G(k) = e^{-|k - \frac{m}{2}|}$.

With the similarity error d_{ij}, the weight is calculated as

$$\theta_{ij} = \frac{1}{C(i)} e^{-\frac{d_{ij}}{h}} \quad (9)$$

where $C(i)$ is the normalizing constant with $C(i) = \sum_j e^{-\frac{d_{ij}}{h}}$. The weights s_{ip} can also be obtained in the same way.

3.2. Discussion

The major features of RLLR are highlighted as follows: (1) high-level temporal and spatial consistency; (2) better preserving temporal motion information and spatial high frequency features; (3) learning with a local linear and global nonlinear manner; (4) easy implementation with a closed-form solution.

4. EXPERIMENTAL RESULTS

In this section, extensive experiment results are presented to evaluate the proposed SI extrapolation scheme in comparison with some state-of-the-art work in the literature. The comparison group includes three other SI extrapolation methods: conventional motion-compensated extrapolation (MCE)[4], MCE with motion vector filtering (MCE+MVF) [5] and two AR model based method [7].

In the experiments, we choose two representative video sequences to demonstrate the efficiency of our method: $Foreman$ featuring moving close-up object and panning background, and $Paris$ featuring fast moving object. In each sequence, 100 frames are selected where Key frames are encoded in H.264 intra mode with three QP values: 20, 24, 28, and WZ frames are encoded with the well-known TDWZ codec [3]. The radius of neighborhood is set to 1 and the radius of training window is 8.

Fig. 2 illustrates the objective quality comparison of SI for $Foreman$ and $Paris$. As depicted in Fig. 2, our method achieves the best SI generation quality compared with the other three methods. The same results are also shown in the overall performance of DVC system, as illustrated in Fig. 3. For $Foreman$ and $Paris$, the gains, compared with the two AR model based method, is up to 0.3dB and 0.8dB, respectively. This is because that in the two video sequences there are heavy irregular and fast motion which can not be well represented using block-based methods. The granularity of our method is pixel level, at which the complex motion can be described more accurately. At the same time, spatial correlation information is also exploited in our scheme, which can bring some advantages for estimating occluded regions.

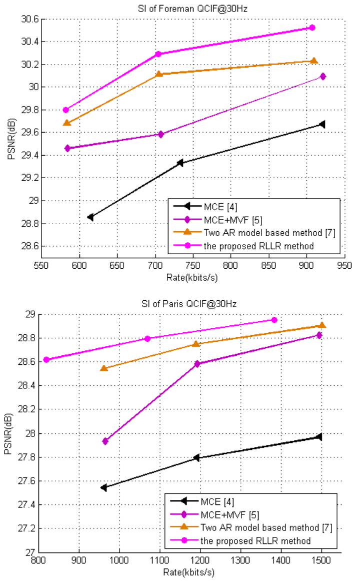

Fig. 2. Comparison of the SI for *Foreman* and *Paris* sequence

5. CONCLUSION

In this paper, we present a novel side information extrapolation scheme for low-delay Wyner-Ziv video coding. Our algorithm is based on a regularized local linear regression model which simultaneously minimizes the moving least squares error on the temporal training samples and preserves the local spatial geometrical structure of the same frame, therefore achieves wonderful temporal and spatial consistency. By fully exploiting the correlation information among spatio-temporal neighboring space, the obtained transformation functions are smooth and locally linear, and can keep the local motion information wonderfully. Experimental results demonstrate the superior performance of our method in comparison with the state-of-the-art methods.

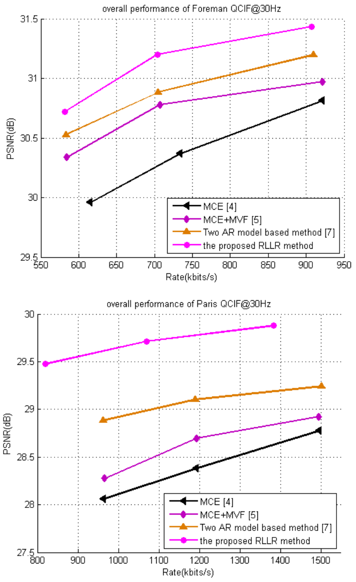

Fig. 3. Comparison of the overall performance for *Foreman* and *Paris* sequence

6. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation 60736043 and 61073083, 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 Trans. Information Theory, Vol.19, pp.471-480, July.1973.

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

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

[4] A. Aaron, E. Setton and B. Girod, “Towards practical Wyner-Ziv coding of video”, Proc. IEEE International Conference on Image Processing , ICIP-2003, Barcelona, Spain, Sept. 2003.

[5] L. Natrio, C. Brites, J. Ascenso, F. Pereira, “Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding,” Int. Workshop on Very Low Bitrate Video Coding, Sardinia, Italy, September 2005.

[6] S. Borchert, R. P. Westerlaken, R. K. Gunnewiek, and R. L. Lagendijk, “On extrapolating side information in distributed video coding,” Proceedings of Picture Coding Symposium, Lisbon, Portugal, November 2007.

[7] Y. Zhang, D. Zhao, S. Ma, R. Wang, and W. Gao, “An autoregressive model for improved low-delay distributed video coding,” Proc. SPIE Visual Communications and Image Processing, California, USA, Jan.18-22, 2009.

[8] D. Levin. “The approximation power of moving least squares,” Mathematics of Computation, 67(224):1517-1531, 1998.

[9] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” Multiscale Modeling Simulation, vol. 4,no. 2, pp. 490-530, 2005.