A NOVEL SCALABLE TEXTURE VIDEO CODING SCHEME WITH GPCA

Jian Liu,Yueting Zhuang, Lei Yao, Fei Wu

College of Computer Science, Zhejiang University
Hangzhou, 310027, China

ABSTRACT

This paper proposes a novel SNR scalable coding method with the support of generalized principle component analysis (GPCA). This method encodes the low-pass and high-pass pictures generated by the MCTF decomposition with a hybrid linear model instead of traditional block-based DCT transform. GPCA is a powerful tool to identify the hybrid linear model in the textures, which segment the texture into heterogeneous regions, and then encode each region with PCA method. By keeping various proportions of PCA coefficients, and altering the quantization step sizes for different layers, a better scalable coding result can be achieved.

Index Terms— Video coding, scalability, texture coding, GPCA, hybrid linear model

1. INTRODUCTION

Scalable video coding is to encode a video sequence into a stream with multiple embedded sub-streams, with each of the sub-streams being a compression of the original video sequence at a particular resolution. Such a stream can accommodate various network bandwidths and different user requirements. Undoubtedly this technology is very useful for online video services.

Most of the recent industrial video coding standards include scalable coding methods. The fine granularity scalability (FGS) approach included in ISO/IEC MPEG4 standard provides SNR (Signal to Noise Rate, or quality) scalability by re-quantizing coefficients of the discrete cosine transform (DCT) [1]. In January 2005, MPEG and ITU-T decided to include the scalable extension of H.264/AVC [2][3] as an amendment of the standard, which has achieved the best ever compression efficiency. The joint video team (JVT) of ITU-T is still working on the project. The scalable extension of H.264/AVC uses motion-compensated temporal filtering (MCTF) instead of the closed-loop motion compensated prediction structure to implement temporal scalability. For SNR scalability, it uses a similar FGS approach as in MPEG4, except that the high quality references are employed to improve the coding efficiency.

The above mentioned SNR scalability implementations are all constructed on block-based DCT transform. However, recent studies of GPCA in image processing [4] gave us the inspiration that a global sparse representation for the pictures in scalable video coding might be a better choice. The reason is that GPCA can exploit the texture differences in various regions of a picture, and encode each texture separately. This feature is not only useful in improving the coding efficiency, but may also be promising in enabling the end user to access particular sub-streams of a video sequence that represents a certain texture area. Therefore we come up with a GPCA-based SNR scalability scheme. This scheme is designed on the MCTF coding structure. A video sequence is turned into a series of low-pass and high-pass pictures after MCTF. These pictures are processed by GPCA to be segmented into a variety of regions, each of which can be well represented with one parametric modal. Then these regions are separately coded by a set of PCA bases and coefficients. By discarding insignificant bases and coefficients, we get a coarse representation of the original picture. The discarded bases and coefficients can be used to form different enhancement layers to refine the quality of the decoded video. This scheme provides SNR scalability in fairly fine granularity. Experiments show that this approach typically gets a better PSNR than DCT based method for most test sequences.

2. SCALABLE TEXTURE CODING WITH GPCA

2.1. Overview

The proposed scalable video coding scheme achieves temporal and spatial scalability using the current scalable extension of H.264/AVC test model JSVM 3.0 [5]. For SNR scalability we use a hybrid linear model representation in place of DCT transform to encode the high-pass and low-pass pictures generated by MCTF, which we call textures. The framework is shown in Figure 1.

2.2. Spatial Scalability

As shown in Figure 1, the video source is decimated by a factor of 2, both vertically and horizontally, to obtain a spatial base layer. The two spatial layers are then separately

encoded with the same procedure, except that four kinds of inter-layer prediction are used between them.

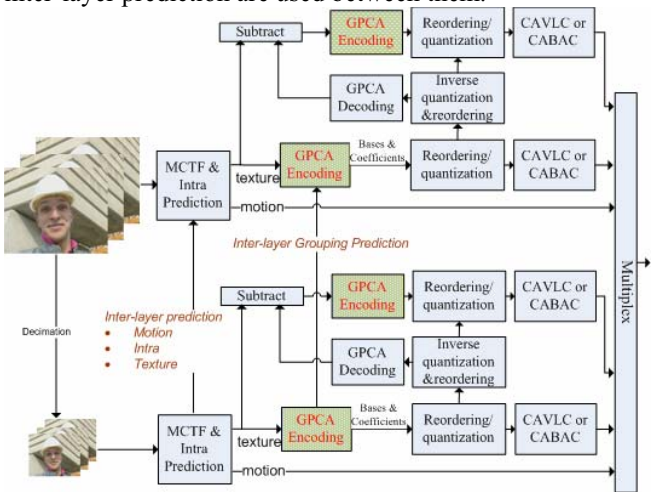

Figure 1. Overall framework of the GPCA-based scalable texture video coding scheme.

1. Scale up the motion vectors of the base layer to predict the motion vectors of the enhancement layer.
2. Up-sample the macro-blocks of the base layer to predict the macro-blocks of the enhancement layer.
3. Up-sample the textures of the base layer to predict the textures of the enhancement layer.
4. Scale up the grouping (segmentation of vectors in the GPCA method) of the base layer to predict the grouping of the enhancement layer.

The first three kinds of prediction are from the JSVM test model [5]. The forth one is for the GPCA coding method, which will be described in the next section. Of course all the four kinds of inter-layer prediction can be switched on and off adaptively.

2.3. Temporal Scalability

For temporal scalability, our framework employs the MCTF extension of H.264/AVC. MCTF is a wavelet lifting method which decomposes a video sequence into several sets of low-pass and high-pass pictures. The low-pass pictures form a temporal base layer, while each set of high-pass pictures form a temporal enhancement layer. For the details of MCTF, please refer to [6].

2.4. SNR Scalability with GPCA

The proposed SNR scalability includes coarse granularity which is realized through re-quantization, and fine granularity which is realized by discarding GPCA data components. In the scalable extension of H.264/AVC, the textures, i.e. high-pass and low-pass pictures, are coded using DCT based integer transform. DCT converts a picture into the frequency domain, and represents it with a superstition of basic functions. This set of basic functions is invariant for every picture. This method certainly does not take account of the fact that an image typically contains regions of different textures. A recent work by René Vidal, Yi Ma et al. [7] called GPCA, can simultaneously segment an image into different regions and approximate each region with a linear model. This method, also called hybrid linear model representation of images, is introduced into video texture coding in our work.

Hybrid Linear Model for Texture Encoding

To apply GPCA on video texture coding, we first divide each texture picture into $l \times m$ blocks. Assume that every pixel has 3 color components: luma, Cb and Cr. For each block, the luma values of every pixel are stacked into a D dimensional vector $v \in \mathbb{R}^D$, where $D = l \times m$. Likewise, Cb and Cr values of each block are also stacked into D dimensional vectors. Then all the vectors reside in a D dimensional space $\mathbb{R}^D$. Figure 2 shows the construction process of vectors in a YUV420 format video frame. For a picture with width W and height H, the total number of luma vectors extracted is $N = (W \times H)/(l \times m)$. As the number of Cb samples is 1/4 the number of luma samples, the number of Cb vectors is $N/4$. The same condition holds for Cr vectors. For other strategies of mapping frames to vector spaces, please refer to [8].

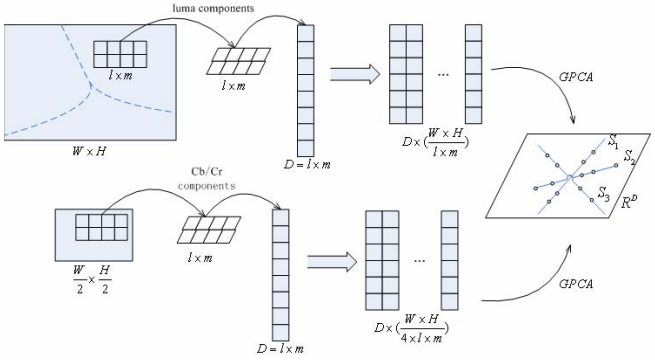

Figure 2. Converting a texture frame into a set of vectors, which are then segmented and estimated by GPCA.

Now given a set of vectors, GPCA will identify heterogeneous groups and approximate each group with a linear model. Assume that the luma vectors can be segmented into n groups $G_i|_{i=1}^n$. Every group forms a subspace S_i of the D dimensional ambient space $\mathbb{R}^D$. For each S_i, a basis $B_i = \{b_{ij}|_{j=0}^{k_i}\}$ can be found. Then a given vector $v \in G_i$ can be represented as $\sum_{j=0}^{k_i} a_j b_{ij}$, where a_j are the coefficients.

A more detailed description of the algorithm is presented below. At the beginning, an initial pass of PCA is performed to reduce the dimension of the ambient space. The vector set $\{v_i \in \mathbb{R}^D\}_{i=1}^N$ is subtracted by the mean vector $\overline{v} = \frac{1}{N}\sum_{i=1}^N v_i$, resulting in a vector set which has a zero mean. The set $\Delta V = \{v_i - \overline{v}\}_{i=1}^N$ can be represented by its

SVD decomposition $\Delta V = USV^T$, and the first d columns of U become the projection base $P \in \mathbb{R}^{D \times d}$. The reduced dimension d and the PCA coefficients C are computed as follows:

$$S = diagonal(\alpha_1, \alpha_2, \ldots, \alpha_D)$$
$$d = \min_{k=1,\ldots,D-1} \left(\sum_{i=k+1}^{D} \alpha_i^2 < \varepsilon \right) \quad (1)$$
$$C = S(1:d, 1:d)V^T(1:d, :)$$

Now the original vectors are represented by the set of coefficients C, a second pass of GPCA is performed on C. C is further segmented into n groups $\{g_i \in \mathbb{R}^{d \times n_i}\}_{i=1}^{n}$, where n_i is the number of vectors that belong to group g_i. Each group g_i is then represented by a set of bases $b_i \in \mathbb{R}^{sd_i \times d}$, where sd_i is the dimension of the subspace that g_i belongs to, and a set of coefficients $C_i \in \mathbb{R}^{sd_i \times n_i}$ similar to that in the first pass of PCA. Now a texture frame is converted to a set of coefficients:

$$Frame = P + \{C_i + b_i + meanVector_i\} + meanVector \quad (2)$$

The total number of coefficients for encoding the texture is:

$$Count = d \times D + \sum_{i=1}^{n} (sd_i \times (n_i + d) + d) + D \quad (3)$$

For more information on GPCA, please refer to [7].

Reordering of Data Elements

Once the texture frame has been transformed into a set of bases and coefficients, we design a reordering procedure to organize these data elements into a suitable format for transmission.

Here is a list of data elements obtained through GPCA encoding: P (bases of the first pass PCA); M (mean vectors of the first pass PCA); $m_i|_{i=1}^{n}$ (mean vectors of the n groups in the second pass GPCA); $b_i|_{i=1}^{n}$ (bases of the n groups in the second pass GPCA); $C_i|_{i=1}^{n}$ (coefficients of the n groups in the second pass GPCA). Note that the above data can all be divided into luma, Cb and Cr partitions. From the dimensions of these data elements, it is easy to see that $C_i \in \mathbb{R}^{sd_i \times n_i}$ account for a major proportion of the data.

In the above list, P, M and m_i are indispensable and so are transmitted first. For b_i and C_i, we have the observation illustrated in Figure 3. In each color channel we get 2 subspaces g_1 and g_2. Take the g_1 subspace in the luma channel for example, the first basis vector and the first coefficient vector are both shown in grey color. The dimension of g_1 is sd_1, so there are sd_1 basis vectors. There are n_1 vectors in this subspace, so there are n_1 columns of coefficients. At the reconstruction stage, the first row of the coefficients in the figure will multiply the first basis vector. So if only the first basis vector (shown by the grey bar) is lost, the first row of coefficients is useless. This condition holds for all other rows. Note another fact that the bases are ordered by their significance. So the first basis vector is most significant. Thus, we transmit the bases and coefficients in the order shown by the dotted arrow line and the numbers on the left. The first row of each subspace will be coded first, then the second row, and so on so forth.

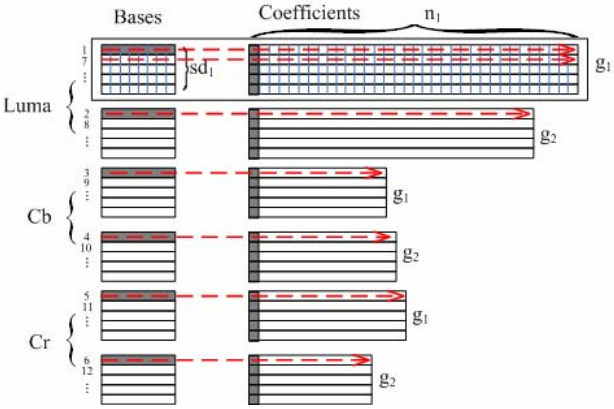

Figure 3. Reordering of data elements in order to implement fine granularity scalability.

By reordering the data components obtained by GPCA coding, the bases and coefficients can be truncated at a wide range of points. If the data is truncated, some insignificant bases are discarded, or part of the coefficients for a basis vector is set to zero, which results in a degraded texture frame. In this way, we realized fine granularity scalability.

We didn't use the H.264/AVC NAL unit syntax for the coding of GPCA bases and coefficients. Instead we defined a temporal data format.

3. EXPERIMENTAL RESULTS

We set up an experiment to evaluate the performance of our GPCA-based scalable texture video coding scheme. The experiment is done on the scalable H.264/AVC extension test model JSVM 3.0. We embed our GPCA image coding method into the test model to encode the high-pass and low-pass pictures. The resulting bases and coefficients are reordered using the technique described in the last section and partially discarded to evaluate the fine granularity SNR scalability performance of this scheme.

Figure 4 shows the comparison between GPCA-based SNR scalable coding and DCT-based SNR scalable coding. The curves demonstrate the average PSNR of the second layer for the video sequence "vectra". The solid curve are obtained by using 4×4 block based DCT for texture coding. We can see that the average PSNR attained with GPCA method are generally better than that of DCT. Furthermore, when gradually truncating enhancement layer data elements, the curves of GPCA performance decline smoothly. This feature promises a smooth change of the video quality when the network transmission varies. The test sequence "vectra"

is one with dramatic background motion. We also tested our algorithm on other sequences, which got us very similar results.

In another case, we changed coding parameters to test the impact of inter-layer prediction on coding performance. Table 1 presents the results of several video sequences encoded with different options. We can see that both the inter layer prediction and grouping prediction cause minor PSNR loss. However this is acceptable for the considerable reduction of complexity.

option / seq.	(1)	(2)	(3)	(4)
vectra	Y:38.3915 U:43.0975 V:43.3396	Y:38.3922 U:43.1081 V:43.3554	Y:38.8399 U:43.2827 V:43.4470	Y:38.8442 U:43.3068 V:43.4731
foreman	Y:38.0424 U:42.2288 V:45.2592	Y:38.0428 U:42.2413 V:45.2604	Y:38.3435 U:42.9812 V:45.1166	Y:38.3489 U:42.9979 V:45.1162
news	Y:38.3670 U:41.4038 V:42.3655	Y:38.3682 U:41.4022 V:42.3620	Y:38.5663 U:41.5221 V:42.4022	Y:38.6053 U:41.5312 V:42.4251
football	Y:37.7125 U:42.5596 V:43.2864	Y:37.7235 U:42.5888 V:43.3186	Y:38.3423 U:42.7167 V:43.2687	Y:38.3544 U:42.7516 V:43.3065
stephan	Y:36.6015 U:39.9481 V:40.2245	Y:36.6012 U:39.9513 V:40.2351	Y:37.2878 U:40.5836 V:40.7578	Y:37.2893 U:40.5903 V:40.7653

Table 1: PSNR value of different video sequences encoded with different options. (1) both inter layer prediction and grouping prediction are on; (2) inter layer prediction is on while grouping prediction is off; (3) inter layer prediction is off while grouping prediction is on; (4) neither inter layer prediction nor grouping prediction is off.

4. CONCLUSION

We introduced hybrid linear model into the field of video texture coding. By replacing DCT transform with GPCA segmentation and modeling, and by carefully reordering the bases and coefficients obtained by GPCA, a new SNR scalable coding scheme is proposed. Experiments show that this scalable coding approach gets comparable or better results than DCT based methods. The computational complexity problem has not been paid enough attention, but a grouping prediction method is raised to reduce the complexity of the algorithm. The segmentation and estimation ability of GPCA approach is very promising and may also have other applications in video coding. The future work can be about network adaptation of the bit stream, or organizing different texture regions in separate layers to enable a new kind of scalability.

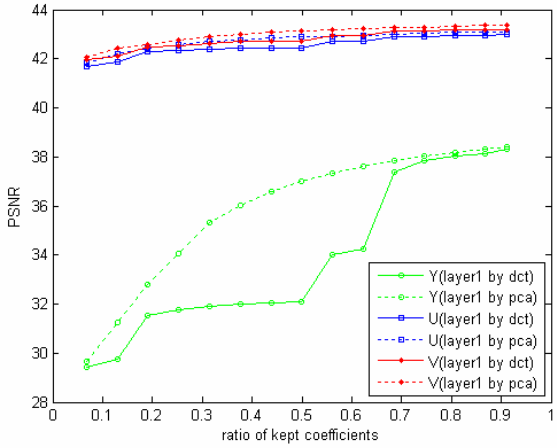

Figure 4. Comparison of GPCA and DCT texture coding methods with the test sequence vectra.

Acknowledgments: This work is supported by National Natural Science Foundation of China (No.60533090, No.60525108), Science and Technology Project of Zhejiang Province (2005C13032，2005C11001-05)，and China-US Million Book Digital Library Project(www.cadal.zju.edu.cn)

5. REFERENCES

[1] ISO/IEC 14496-2, “MPEG-4 video FGS v.4.0.,” Tech. Rep. N3317, Proposed Draft Amendment (PDAM), Noordwijkerhout, the Netherlands, March 2000.

[2] Heiko Schwarz, Detlev Marpe, and Thomas Wiegands, “Overview of the Scalable H.264/MPEG4-AVC Extension,” to be published in proceedings of *ICIP 2006’*.

[3] Heiko Schwarz, Detlev Marpe, Thomas Schierl, and Thomas Wiegand, “Combined scalability support for the scalable extension of H.264/AVC,” IEEE International Conference on Multimedia and Expo, 2005. 10.1109/ICME.2005.1521456

[4] Kun Huang, Allen Y. Yang, and Yi Ma, “SPARSE REPRESENTATION OF IMAGES WITH HYBRID LINEAR MODELS,” *ICIP 2004’*.

[5] JVT-P202, “Joint Scalable Video Model JSVM-3”.

[6] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “MCTF AND SCALABILITY EXTENSION OF H.264/AVC,” *ICIP 2004’*.

[7] Ren’e Vidal, Yi Ma, Shankar Sastry, “Generalized Principal Component Analysis (GPCA) ,” Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’03) 1063-6919/03, 2003 IEEE.

[8] L. Yao, J. Liu, and J.Q. Wu, “An Approach to the Compression of Residual Data with GPCA in Visual Coding,” *PCM 2006’*.