Multimed Tools Appl (2009) 41:111–123
DOI 10.1007/s11042-008-0221-4

The training of Karhunen–Loève transform matrix and its application for H.264 intra coding

Yi Gao · Jiazhong Chen · Shengsheng Yu · Jingli Zhou · Lai-Man Po

Published online: 17 September 2008

Abstract In H.264/AVC, 4 × 4 discrete cosine transform (DCT) is performed on the residual signals after intra prediction for decorrelation. Actually, residual blocks with different prediction modes exhibit different frequency characteristics. Therefore, the fixed transform matrix cannot match the energetic distribution of residual signals very well, which degrades the decorrelation performance. Fortunately, the energetic distributions of residual blocks with the same mode are relatively coincident, which makes it possible to train a universally good Karhunen–Loève transform (KLT) matrix for each mode. In this paper, an optimal frequency matching (OFM) algorithm is proposed to train KLT matrices for residual blocks and nine KLT matrices corresponding to nine prediction modes of 4 × 4 intra blocks are trained. Experimental results show that KLT with trained matrices yields a persistent gain over H.264 using 4 × 4 DCT with an average peak signal-to-noise ratio (PSNR) enhancement of 0.22dB and a maximum enhancement of 0.33dB.

Keywords Karhunen–Loève transform · Discrete cosine transform · Intra coding · H.264/AVC

1 Introduction

The H.264/MPEG-4 part 10 is the latest video coding standard of the Joint Video Team (JVT), formed by ISO/IEC MPEG and ITU-T VCEG [20]. Its performance is superior to other well-known video coding standards such as H.263 [7], MPEG-4 [16], in terms of peak signal-to-noise ratio (PSNR) at the same bit rate [19]. Its high coding efficiency is made possible by new advanced coding tools such as variable block size motion estimation (ME), multiple reference frames, quarter-pixel accuracy ME, and spatial prediction for intra

Y. Gao · J. Chen · S. Yu (✉) · J. Zhou
Department of Computer Science, Huazhong University of Science & Technology,
Wuhan 430074, China
e-mail: ssyu@mail.hust.edu.cn

L.-M. Po
Department of Electronic Engineering, City University of Hong Kong, Kowloon,
Hong Kong 999077, China
e-mail: eelmpo@cityu.edu.hk

coding [17]. Usually, inter coding is superior to intra coding, but intra coding is useful for various purposes such as random access, video editing, and scene extracting [10].

In H.264, intra prediction is conducted in the spatial domain, by referring to neighboring samples of previously decoded blocks that are to the left and/or above the block to be predicted. In all slice-coding types, two primary types of Intra coding are supported: intra 4 × 4 and intra 16 × 16 prediction. There are nine 4 × 4 and four 16 × 16 modes to be used as the candidates for the best mode. The intra 4 × 4 mode is based on predicting each 4 × 4 block separately and is well suited for coding the picture regions with significant detail. The intra 16 × 16 mode, on the other hand, does prediction and residual coding on the entire 16 × 16 block and is more suitable for coding very smooth areas [11].

The key in efficient image and video compression is to explore source correlation so as to find a compact representation of image and video data [8]. Over the past few decades, various spatial transforms, such as the Karhunen–Lòeve transform (KLT), discrete cosine transform (DCT), and discrete wavelet transform (DWT) [18], have been developed to explore source correlation. The H.264/AVC performs DCT on the residual signals after a spatial prediction for energy compaction. As indicated in [9], residual blocks with different optimal prediction modes exhibit different frequency characteristics, thus the fixed DCT matrix cannot match the frequency characteristics of residual signals very well. Therefore, the decorrelation performance of DCT is less satisfactory. In this work, we turn to KLT, of which the matrix is trainable, for better decorrelation performance.

The KLT is known as a data-dependent transform that achieves optimal decorrelation (all off-diagonal terms of the transformed data's covariance matrix are identically equal to zero) and optimal energy compaction [5]. It is preferable to DCT and DWT in some applications in order to achieve higher compression efficiency [12–14]. However, in these applications, the trained KLT matrices were optimal only with respect to a strict class of signals. In other words, KLT would have to either be chosen in advance (which means that the statistics of the data have to be known in advance) or computed during the encoding process and communicated to the decoder, which is expensive in terms of both computational complexity and rate [3]. The performance penalty associated with using a worst KLT rather than a best KLT can be made arbitrarily large in both the fixed-rate and the variable-rate transform coding scenarios [4].

Therefore, our research problem in this work becomes: can we train universally good KLT matrices so that the KLT transform do not need to adapt its transform matrix according to input residuals? Fortunately, the energetic distributions of residual blocks with the same mode are relatively coincident [17], which makes us possible to train a universally good KLT matrix for each mode. In this paper, an optimal frequency matching (OFM) algorithm is proposed to train universally good matrices. In this algorithm, after dividing samples (in the rest of this paper, the actual input signal is named as sample while the possible input as signal) into groups, a great number of candidate KLT matrices can be obtained using conventional method. Due to the relatively coincident energy distribution of residual signals with the same prediction mode, the frequency spectrum of selected residual samples can be approximately taken as that of the residual signals, which will be then referred to select the best matrix from all candidate KLT matrices. Additionally, in order to make decorrelation performance of the final selected matrix as good as possible, we permute each candidate KLT matrix. Finally, the permutated KLT matrix that can absorb the most energy of signals into the lower bands is selected as output. Using this method, we have trained nine KLT matrices corresponding to the nine prediction modes for 4 × 4 intra blocks. Experimental results demonstrate that the decorrelation performance of KLT with these matrices is much better than DCT persistently.

The remainder of the paper is organized as follows. In the section 2, the definition of KLT is first reviewed and thought way of OFM algorithm is then introduced. The details of OFM algorithm are described in the section 3. In the section 4, experiments and analysis of our KLT matrices with DCT are illustrated. Finally, concluding remarks are given in the section 5.

2 Overview and motivation

First of all, we will review the definition of KLT that is an orthogonal transform. Let vectors V_1, V_2,...,V_L to be L vectors of length M. The average vector φ and each difference vector D_i are calculated as following, respectively:

$$\varphi = \frac{1}{L}\sum_{i=1}^{L} V_i \tag{1}$$

$$D_i = V_i - \varphi. \tag{2}$$

The difference vectors are used to set up covariance matrix C as

$$C = \frac{1}{L}\sum_{i=1}^{L} \left(D_i^T D_i\right). \tag{3}$$

Then the KLT of V_i is defined by

$$Y_i = (V_i - \varphi)\Omega \tag{4}$$

where $\Omega = [\Omega_1, \Omega_2, ..., \Omega_M]$is the KLT matrix and Ω_i is the transpose of the i-th eigenvector of C. For linear prediction like intra prediction in H.264, the prediction residual signals can be assumed to be Gaussian white noise with zero mean. Thus, (4) can be simplified as

$$Y_i = V_i\Omega. \tag{5}$$

Based on (5), we can obtain

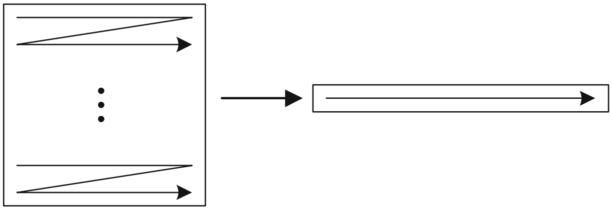
$$Y_i^k = V_i\Omega_k \tag{6}$$

where Y_i^k represents the k-th element of Y_i. Both perfect reconstruction and orthogonality can be sufficed according to the definition of KLT. For block transform coding, we can easily formulate a vector by performing raster scan on this block, as illustrated in Fig. 1. After quantization, the transform coefficients are sequentially scanned for entropy coding.

We can take KLT as horizontal multi-channel wavelets transform (MCWT) [15] with equal filter length and channel number. Accordingly, each basis vector of a KLT matrix can be viewed as a channel. In transform coding, the multi-channel wavelets with good transform performance are expected to have good frequency property in order to compact more energy into the lower frequency bands [2].

From the above analysis, a universal KLT matrix must also be multi-channel wavelets with good frequency property. Actually, it is hard to exploit the relationship between training samples and the frequency property of the trained KLT matrix. That is to say we are unable to formulate principles for sample selection to guarantee the good frequency property of the KLT matrix. Nevertheless, it is much more likely to obtain a good KLT matrix from a KLT matrix candidate set when the number of candidates is large enough. Therefore, our research problem becomes: can we design an algorithm to find the KLT matrix with optimal frequency property from a great number of KLT matrices? We propose an OFM algorithm, which will be elaborated in the following sections.

Fig. 1 Input data block realignment



3 OFM algorithm for KLT matrix training

As indicated earlier, a good KLT matrix is expected to possess good frequency property so that most energy can be compacted into the lower bands. In finding the optimal KLT matrix, there are two important issues that need to be illuminated:

1) How can the universally good decorrelation performance of the final selected KLT matrix be guaranteed?
2) Since good frequency property is quite fuzzy, is there any way to describe good frequency property for a 1-D transform quantificationally, which allows us to find the optimal matrix using a dynamic programming solution?

For the first question, it must be emphasized that our goal is to find the KLT matrix that has optimal performance on the whole. That means its optimal performance for individuals is not ensured. Moreover, by "universally good," we mean the good performance for various video sequences. In this case, the frequency spectrum of a universally good KLT matrix is expected to match statistical characteristics of signals well. Since the energy distributions of residual signals with the same intra prediction mode in H.264 are relatively coincident, for each mode, it is reasonable for us to approximately extract the statistical characteristics of residual signals through that of a certain amount of residual samples.

Suppose that, there are totally N sample vectors (namely, the realigned residual blocks). If we connect n of them ($n \leq N$) end to end to form a new vector, the frequency spectrum of this newly formed vector can reflect the energy distribution of these samples. For convenience, we call this newly formed vector *string* vector. Figure 2 illustrates the construction of a *string* vector from n vectors of length M. Therefore, we can take the frequency spectrum of this *string* vector as the energy distribution of signals. Clearly, to reduce the influence of individual differences to the frequency spectrum of the *string* vector, n should be large enough. Therefore, a universally good KLT matrix is expected to have good decorrelation performance to the *string* vector.

Since KLT can be taken as MCWT, (6) can be viewed as a convolution operation. As convolution in the time domain is equivalent to multiplication in the frequency domain [6], The energy distribution of the transform coefficient Y_i^k can be acquired by multiplying the frequency spectrum of the *string* vector and that of Ω_k. By calculating the proportion of energy of transform coefficients in lower bands (with smaller k) to the whole frequency band, we can judge whether this KLT matrix yields good decorrelation performance to this *string* vector or not. For the i-th channel of a KLT matrix X, its transform gain $g_X(i)$ can be calculated according to the energy distribution proportion of this channel. Considering that transform coefficients of different frequency bands have different influence to entropy coding, the total transform gain of X that is defined as $g(X)$ should be calculated

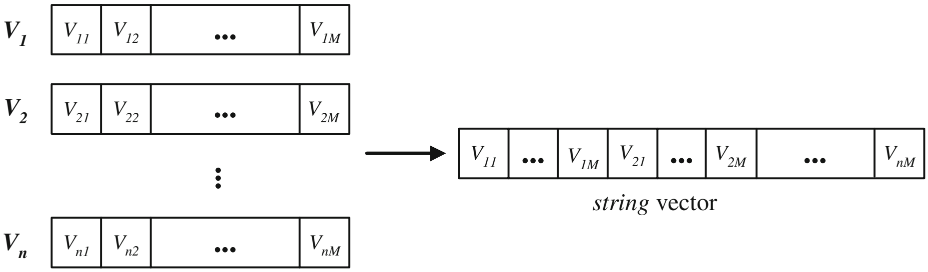


Fig. 2 Construction of a *string* vector

as a weighted sum of gain of each channel. Therefore, $g_X(i)$ and $g(X)$ can be defined as following:

$$g_X(i) = \frac{e_i(X)}{\sum_{j=1}^{M} e_j(X)} \tag{7}$$

$$g(X) = \sum_{i=1}^{M} w_i \times g_X(i) \tag{8}$$

where $e_i(X)$ is the energy distributed in the i-th band and w_i is the weighting coefficient of the i-th band. The bigger the gain of a KLT matrix is, the better its performance is.

In (8), with i increased, w_i decreases but the change of $g_X(i)$ is uncertain. Additionally, $g_X(i)$ may be suboptimal under current order of channels. With these two drawbacks, it seems hard to get a high value of $g(X)$. Given a permutation matrix P, it can be demonstrated that $P\times X$ is still orthogonal and perfect reconstruction can be also sufficed. By permutation, both of these two drawbacks may be overcome. That means for a given KLT matrix, a higher transform gain may be achieved by adjust order of its channels.

For an $M\times M$ KLT matrix, there are totally $\prod_{i=1}^{M} i$ permutations. Obviously, the globally optimal way to find the best matrix is to try all permutations. However, this is of high complexity: a 16 × 16 KLT matrix can be permuted in (16!) = 2,004,189,184 ways! So we turn to suboptimal but efficient algorithms. Since the goal of transformation is to compact more energy into lower frequency bands, we can conclude that the lower bands are more important. Accordingly, we should give top-priority to the channels with smaller indexes in permutation. In Fig. 3, the solid curve is the frequency spectrum of a *string* vector. Suppose the KLT matrix has four channels. We then divide the whole frequency range into four bands uniformly, which are partitioned by dash lines and denoted by ①, ②, ③ and ④ respectively. As can be seen, ① possesses the most energy, and then ③ and ②. The least energy is distributed in ④. So we first select from 4 basis vectors the basis vector that can absorb the most energy from this *string* vector in ① to be channel 1. Afterwards, among the remaining 3 basis vectors, the vector that can absorb the most energy in ③ will be selected to be channel 2. Using the same method, the remaining 2 channels can also be decided. Therefore, for an $M\times M$ KLT matrix, the total number of permutations T(M) is:

$$T(M) = \sum_{i=1}^{M} i = \frac{M \times (M+1)}{2} << \prod_{i=1}^{M} i\cdot \tag{9}$$

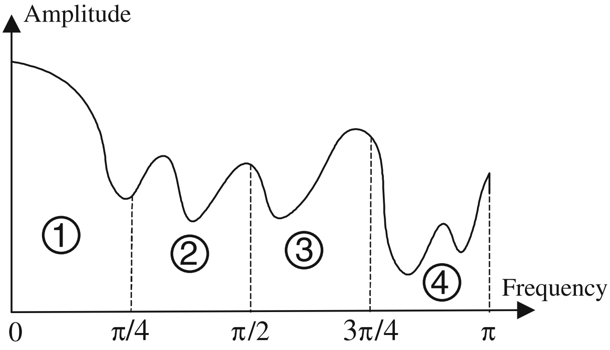


Fig. 3 Frequency band partition in OFM algorithm

Figure 4 illustrates the procedure of OFM algorithm. We first construct a *string* vector by a large number of samples to extract energy distribution of signals. Then a KLT matrix candidate set is obtained by training the grouped samples. For a candidate KLT matrix, its transform gain can be calculated according to its efficiency on compacting signal energy into lower bands. If its transform gain is bigger than that of the trained optimal KLT matrix, this KLT matrix will be regarded as the new optimal matrix. Otherwise, the optimal matrix remains unchanged. After all candidate KLT matrices are considered, the optimal matrix will be taken as final output.

A detailed description of OFM algorithm follows.

OFM algorithm
Initialize:
    max_gain=0.
    Set A to be a unitary matrix.
    Set SVS = $\{s_1, s_2, ..., s_N\}$, where s_i stands for a sample vector.
    For i=1 to N
        Calculate KLT matrix C based on L sample vectors $s_i, s_{i+1}, ..., s_{i+L-1}$.
        $i = i + L$.
        Set X to be a zero matrix.
        Set basis vectors set BVS to be *C*.
        For j=1 to M
            Select best vector v_j for the j-th channel of *X* from *BVS*.
            $BVS = BVS - v_j$.
            Calculate $g_X(j)$.
        End
        Calculate g(X).
        If max_gain < *g(X)*
            max_gain = g(X).
            A = X.
        End.
    End.
Output A.

The OFM algorithm is indeed time-consuming. However, it should be noted that such a complex training procedure is fully independent to the encoding process. That means we

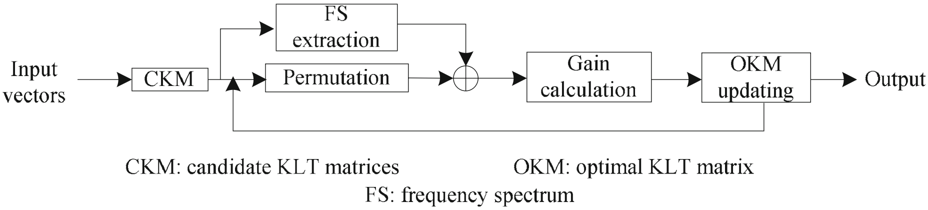


Fig. 4 Flowchart of OFM algorithm

can use the pre-trained KLT matrices instead of training KLT matrices in real time. Moreover, no additional bits are required to code the KLT matrices. Therefore, the KLT transform is very easy to conduct within the H.264 framework. For a 4 × 4 residual block, operation of KLT requires 256 multiplications while DCT requires 128 multiplications (We do not consider additions here). So the computational complexity of KLT is higher than DCT but not too much.

Since the residual blocks with different prediction modes exhibit different frequency characteristics, we use OFM algorithm to train nine KLT matrices corresponding to the nine prediction modes for 4 × 4 intra blocks. In training, we first store a great number of residual blocks of different sequences as training samples. For each mode, we randomly select 100,000 residual blocks of this mode and realign each of them in raster order to be a residual vector. We then connect these residual vectors end to end to form a *string* vector. Afterwards, we take fast Fourier transform to this *string* vector to get its energy distribution characteristics. The frequency spectrum of prediction residuals with vertical mode (mode 0) and horizontal-up mode (mode 8) are illustrated in Fig. 5. The optimal KLT matrix with respect to each mode is finally selected using OFM algorithm. Additionally, considering that coefficients of lower frequency bands are more important to entropy coding, we set the weighting coefficient table $\{w_i\}$ to calculate gain function to be {16, 8, 4, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}.

4 Experiments and analysis

In this section, we examine the coding performance of the proposed 9 KLT matrices in terms of PSNR. For evaluation purposes, the H.264/AVC joint model (JM) software

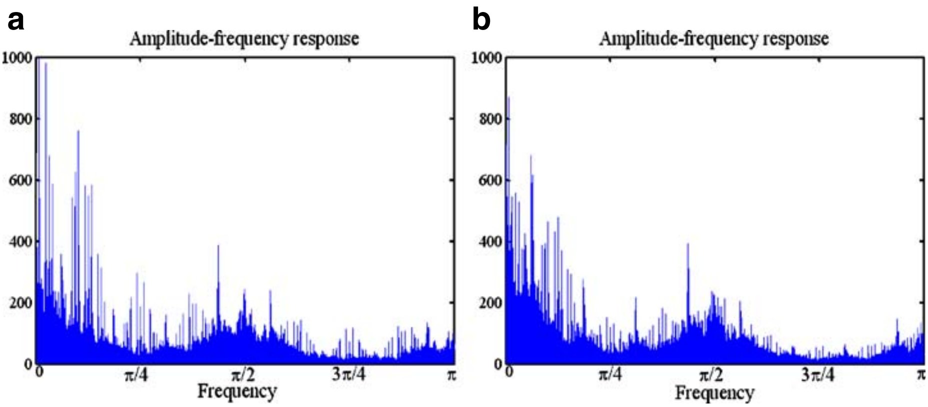


Fig. 5 The frequency spectrum of prediction residuals. **a** mode 0. **b** mode 8

Table 1 Experimental conditions

Coding structure	IIII... (I slice only)
Quantization parameter	20, 24, 28, 32
RD Optimization	Used
Entropy coding	CABAC
8×8 Transform	Unused
Adaptive Rounding	Unused

(version JM10.2) is employed. In the experiments, for each mode, we train an optimal KLT matrix by OFM algorithm and then adopt it to KLT transform for different testing sequences. Experimental results are tested with the conditions as indicated in Table 1. We have conducted an extensive set of experiments with videos representing a wide range of texture and color. Eight sequences in CIF (352 × 288) format were employed: Bus, Container, Coastguard, Foreman, Mobile, Stefan, Tempete and Waterfall. All the video sequences have 100 frames to be encoded at 30 frames per second. To strengthen pertinence of our comparisons, neither adaptive rounding [1] nor 8 × 8 transform is adopted.

The RD curves for both the anchor method (DCT) and the proposed method (KLT) are illustrated in Fig. 6 for the performance comparison. Results of the experiments show that our proposed KLT gives persistent performance gain over the anchor method with an average PSNR enhancement of 0.22dB and a maximum PSNR enhancement of 0.33dB. Moreover, it can be observed that as the bit rate increases, the performance gain also increases. The reason for this is that transform mode affects the coding efficiency differently at different bit rate. A finer quantization process maintains coefficient distribution with more non-zero transform coefficients. Since KLT can compact more energy into lower band than DCT, so the number of non-zero quantized coefficients may be markedly smaller than DCT in this case. On the other hand, regarding the coarse quantization process, since most of the transform coefficients are quantized to zero, transform tools do not make a big difference in the energy distribution. Therefore, similar performance is observed when the bit rate is low, while greater performance improvement is obtained when the bit rate is high.

Another important factor in evaluating the various algorithms is the computational complexity. According to our statistics, the computation time for coding one frame has increased by 10% on average when KLT takes the place of DCT. To be noted, integer DCT is adopted in H.264, which reduces the computational complexity of transform significantly. Consequently, KLT may be more competitive with integer transform implemented.

Figure 7 provides comparisons for energy distribution of the transform coefficients of the first frame of Bus sequence between DCT and our proposed KLT, where QP of 24 is used. It is clear that the transformed coefficients produced by proposed KLT have much less energy than DCT, which means that the proposed KLT matrices possess better decorrelation performance.

5 Conclusions

In H.264/AVC, the transform matrix of DCT is fixed. It cannot match the frequency characteristics of residual signals after intra prediction very well, which degrades the decorrelation performance of DCT. In this study, we turn to KLT, of which the matrix is trainable, for better decorrelation performance. Due to the relatively coincident energetic

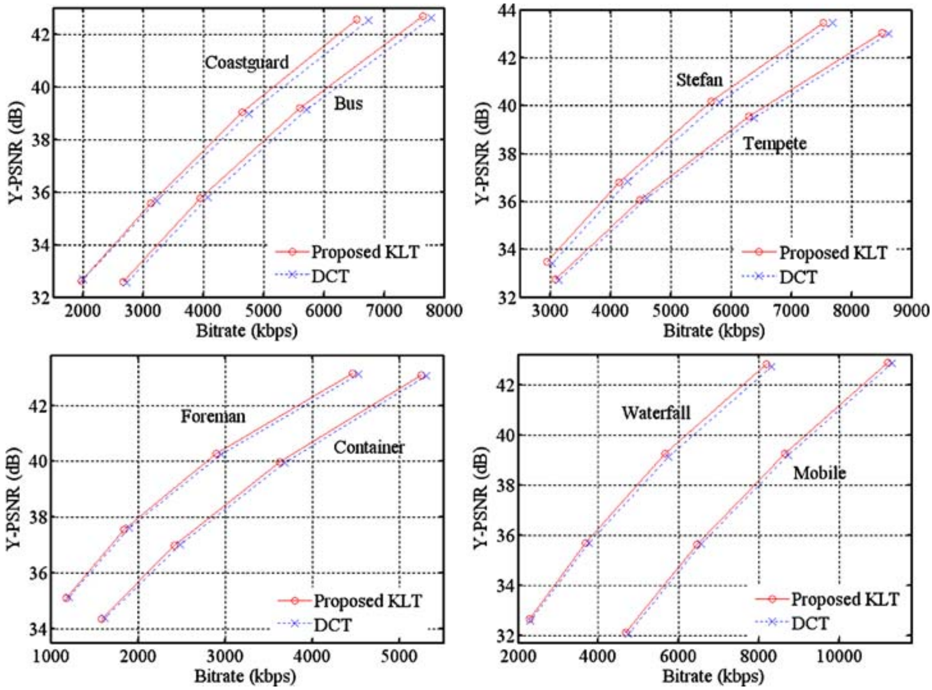
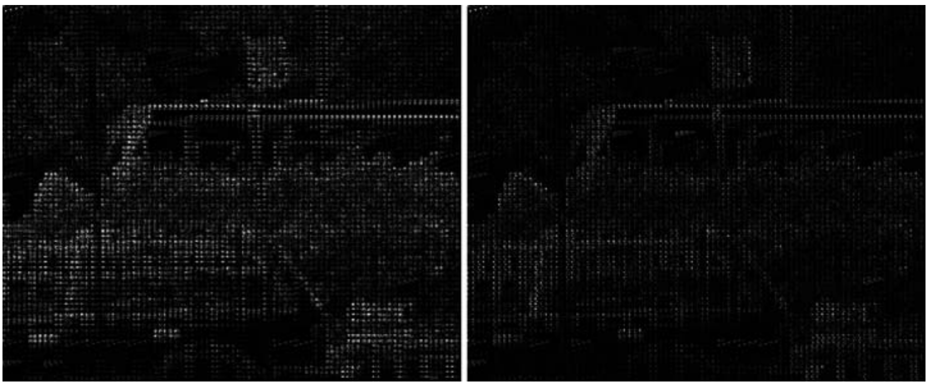


Fig. 6 Comparisons in terms of PSNR for CIF format sequences

distributions of residual signals with the same mode, an optimal frequency matching (OFM) algorithm is proposed to train good KLT matrices. We then use OFM algorithm to train KLT matrices for residual transform for 4×4 intra blocks. Experiments demonstrate that the decorrelation performance of KLT with pre-trained matrices is persistent and much better than DCT. Undoubtedly, OFM can be also used to train KLT matrices of any other size for residual transform. In particular, our proposed OFM algorithm makes KLT practicable for real-time applications under certain circumstance.



a DCT **b** Proposed KLT

Fig. 7 Energy comparisons for transform coefficients of the first frame of Bus sequence

However, there are still several drawbacks in OFM algorithm. First, the residual samples are crucial to matrix selection. Though a large number of training samples are employed to overcome the influence, the performance of the final selected matrix may still be significantly degraded if a considerable number of non-ideal samples (samples whose energetic distributions are very different from that of signals) are included. In our future work, we will consider how to preclude the non-ideal samples to improve the performance of trained matrices. Second, the frequency bands of signals are divided uniformly. It may not be suitable for energy distribution of signals, which will degrade the decorrelation performance of the final selected KLT matrix. Third, how to group samples to get KLT matrix candidate set can be further investigated.

References

1. Adaptive quantization encoding technique using an equal expected-value rule, JVT-N011.doc (2005) http://ftp3.itu.ch/av-arch/jvt-site/2005_01_HongKong, Hong Kong.
2. Chen JZ, Xu J (2006) The application of symmetric orthogonal multiwavelets and prefilter technique for image compression. Multimedia Tools Appl 29(2):175–189. doi:10.1007/s11042-006-0006-6
3. Effros M, Chou PA, Gray RM (1999) Weighted universal image compression. IEEE Trans Image Process 8(10):1317–1329. doi:10.1109/83.791958
4. Effros M, Feng H, Zeger K (2004) Suboptimality of the Karhunen–Loève transform for transform coding. IEEE Trans Inf Theory 50(8):1605–1619. doi:10.1109/TIT.2004.831787
5. Feng HY, Effors M (2004) On the rate-distortion performance and computational efficiency of the Karhunen–Loève transform for lossy data compression. IEEE Trans Image Process 11(2):113–122. doi:10.1109/83.982819
6. Gaydecki P (2001) A versatile real-time deconvolution DSP system implemented using a time domain inverse filter. Meas Sci Technol 12(1):82–88. doi:10.1088/0957-0233/12/1/311
7. Halbach T (2002) Performance comparison: H.26L intra coding versus JPEG2000. JVT-D039
8. He ZH (2007) Peak transform for efficient image representation and compression. IEEE Trans Image Process 16(7):1741–1754. doi:10.1109/TIP.2007.896599
9. Jia J, Jung EK, Kim HK (2007) Adaptive transform coefficient scan for H.264 intra coding. IEICE Trans Inf Syst E 90-D(10):1709–1711. doi:10.1093/ietisy/e90-d.10.1709
10. Lee YL, Han KH, Sim DG, Seo J (2006) Adaptive scanning for H.264 intra coding. ETRI J 28(5):668–671
11. Marpe D, Wiegand T, Sullivan GJ (2006) The H.264/MPEG4 advanced video coding standard and its applications. IEEE Commun Mag 44:134–143. doi:10.1109/MCOM.2006.1678121
12. Martinelli G, Ricotti LP, Marcone G (1993) Neural clustering for optimal KLT image compression. IEEE Trans Signal Process 41(4):1737–1739. doi:10.1109/78.212760
13. Páta P (2006) Suboptimal Karhunen–Loève transform for compression of astronomical images. In: Proc. IEEE International Symposium on Signal Processing and Information Technology, Hawaii, USA, pp 332–337
14. Penna B, Tillo T, Magli E, Olmo G (2007) Hyperspectral image compression employing a model of anomalous pixels. IEEE Trans Geosci Remote Sens Lett 4(4):664–668. doi:10.1109/LGRS.2007.903976
15. Saghizadeh P, Willson AN (1998) A new approach to the design of critically sampled M-Channel uniform band perfect-reconstruction linear-phase FIR filter banks. IEEE Trans Signal Process 46 (6):1544–1577. doi:10.1109/78.678467
16. Sicora T (1997) The MPEG-4 video standard verification model. IEEE Trans Circuits Syst Video Technol 7(1):19–31. doi:10.1109/76.554415
17. Suh KB, Park SM, Cho HJ (2005) An efficient hardware architecture of intra prediction and TQ/IQIT module for H.264 encoder. ETRI J 27(5):511–524
18. Taubman DS, Marcellin MW (2002) JPEG2000: image compression fundamentals, standards and practice. Kluwer, Boston
19. Tsai AC, Paul A, Wang JC, Wang JF (2008) Intensity gradient technique for efficient intra-prediction in H.264/AVC. IEEE Trans Circuits Syst Video Technol 18(5):694–698. doi:10.1109/TCSVT.2008.919113
20. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576. doi:10.1109/TCSVT.2003.815165

Yi Gao was born in 1984. He received the B.S. degree from HUST in 2005, where he is now pursuing his Ph.D. degree in computer science. His main interests are wavelet theory, image processing, and video compression.

Jiazhong Chen was born in 1970. He received the M.S. degree in computation mathematics in 1999 and Ph. D. degree in computer system architecture in 2003 from HUST. He is now an associate professor at School of Computer Science and Technology of HUST. His main research interests include signal processing and wavelet analysis, image and video coding.

Shengsheng Yu was born in 1944, and received the B.E. degree in 1967. He is a Professor and doctor advisor at Huazhong University of Science and Technology. He had been a visiting scholar in west Germany from 1982 to 1983. His main field of research: computer network and storage, discrete signal processing and communication.

Jingli Zhou was born in 1946. She received the B.E. degree in 1969. She is a Professor and doctor advisor at Huazhong University of Science and Technology. She had been a visiting scholar in USA from 1995 to 1996 and has been honor of the State Department Special Allowance since 1999. Her main field of research: computer network and multimedia signal processing.

Lai-Man Po (M'92) received the B.Sc. and Ph.D. degrees, both in electronic engineering, from the City University of Hong Kong in 1988 and 1991, respectively. Since November 1991, he has been with the Department of Electronic Engineering, City University of Hong Kong, and is currently Associate Professor. His research interests are in vector quantization, motion estimation for video compression and H.264/AVC fast mode decision algorithms.