Transform Kernel Selection Strategy for the H.264

Chau-Wai Wong* and Wan-Chi Siu†

Centre for Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
*E-mail: Chauwai.Wong@gmail.com
†E-mail: enwcsiu@polyu.edu.hk

Abstract—**In this paper, we make use of a multiple-kernel scheme as a guide to exploit the transform kernel selection strategy of the hybrid video coding. We have found that the kernel selection tendency lies in a feature which can be extracted from a pair of kernels. We propose using the IK(1,2,1) kernel for I- and P-Frames and the IK(5,7,3) kernel for B-Frames for the H.264 Video Coding Standard. This gives a good improvement in terms of the PSNR and bitrate compared to those using a single kernel in the H.264 (JM 12.2) and the multiple-kernel schemes available in the literature. We also generalize a feature extracted from a pair of kernels to form a feature extracted from a group of many kernels. This collection of corresponding operation points forms the Macroblock-Level Kernel Galaxy (MLKG). Regarding to the robustness and ease of visualizing the performance, we also propose using the MLKG as a guide to design a single-kernel transform coding process for possible future video coding standards.**

I. Introduction

The hybrid video coding has been investigated for more than twenty years and this efficient coding is achieved by making use of both the advantages of predictive coding and transform coding. The Discrete Cosine Transform (DCT) [1][2][3] and subsequently the Integer Cosine Transform (ICT) [4] have been used[1] as the major kernel for transform coding. This is especially true for the H.264 which heavily makes use of the ICT as the transform kernel [5].

In order to improve the coding efficiency, some researchers proposed using a dual kernel system with an alternative sine transform kernel for the H.264 [6][7]. Others argued that a dual kernel system with an alternative DCT-like transform kernel has a better performance [8]. One common point to the approaches is that they used the Rate-Distortion Optimization (RDO) [9] to select the preferred kernel, which is considered as a trial-and-error approach. In this paper, we examine the results generated by the RDO on multiple kernels and further exploit the physics behind. In Section II, we review the multiple-kernel scheme. In Section III, we show the kernel selection tendencies when a multiple-kernel scheme is applied in the H.264 through experimental work. In Section IV, we explain the kernel selection process using a graphical approach. In Section V, we propose a new concept – a feature may be extracted from a pair of kernels and has a crucial effect to the kernel selection process. In Section VI, we make use of the newly-found features and propose using one kernel for each type of frames in the H.264 without employing the multiple-kernel scheme. In Section VII, we further extend the pair of kernels into a group of many kernels to form a Macroblock-Level Kernel Galaxy (MLKG). We show that MLKG is a good tool for testing, grading and eventually determining new kernels for a video coding scheme. Future directions on designing transform coding process are also suggested. In Section VIII, conclusions are drawn.

[1] To some extent, Walsh-Hadamard Transform (WHT) has also been used.

II. Review of the Multiple-Kernel Scheme

The kernels of Integer Cosine Transform mentioned in this paper are denoted as IK(a,b,c) and its complete form is shown in (1). They may also be denoted as (a,b,c) for the sake of simplicity if no ambiguity is found (such as those in TABLE III). Note that the template of integer kernel is orthogonal.

$$\mathbf{H}_{\text{ICT}} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \qquad (1)$$

where a, b and c are all integer values. For example, the H.264 default integer kernel (H.264 Kernel) is denoted as IK(1,2,1) , i.e. $a = 1$, $b = 2$ and $c = 1$ [5].

Let us also simplify the representation of multiple-kernel scheme in the transform coding process, and name it as the Adaptive Kernel Mechanism (AKM) in this paper. The AKM aiming to achieve high coding efficiency is a kernel selection process at the transform coding stage of hybrid video coding. It employs the coding cost calculated by the RDO as the kernel selection criterion. After kernel selection, the encoder will generate a sequence of signaling bits which is the overhead of employing the AKM. In [8], the authors proposed a new DCT-like kernel – IK(5,7,3), and subsequently proved that the AKM comprising of the IK(1,2,1) and the IK(5,7,3) has higher coding efficiency compared to the conventional single kernel arrangement of the H.264. Hence, in this paper, the IK(1,2,1) and the IK(5,7,3) are used for investigations.

III. Kernel Selection Tendencies for Different Types of Frames

After employing the AKM, we further examined the results by the types of frames, and found that there exist different kernel selection tendencies for different types of frames. As it is shown in TABLE I, most macroblocks of I- and P-Frames are selected to be coded with the IK(1,2,1), while most macroblocks of B-Frames are selected to be coded with the IK(5,7,3). The overwhelming kernels are shown in the table and their counterparts are omitted since they can obviously be deduced. We can see that for all sequences, the results are mainly uniform. Note that TABLE I has only shown the kernel selection tendency at *QP* = 27. Nevertheless, as *QP* increases, although the selection bias is not so obvious as it is shown in TABLE I, the bias still exists.

TABLE I

Kernel Selection Tendencies for I-, P- and B-Frames, *QP* = 27

Sequence		Percentage (%)		
Size	Name	I-Frame coded with IK(1,2,1)	P-Frame coded with IK(1,2,1)	B-Frame coded with IK(5,7,3)
QCIF	Container	80.49	96.05	85.04
	Foreman	92.71	94.25	97.24
	Silent	97.92	94.99	96.26
CIF	Paris	93.63	88.24	99.41
	Foreman	85.28	94.30	95.50
	Mobile	98.20	96.96	99.80
	Tempete	98.21	95.12	99.48
HD (720p)	BigShips	87.80	97.32	73.38
	City	93.48	97.79	85.04
	Crew	73.01	90.49	89.13
	Night	87.29	92.29	98.80
	ShuttleStart	85.35	97.33	74.07

The above observation severely conflicts with our "common understanding" on the relationship among I-, P- and B-Frames and the property of the DCT kernel. On one hand, as the prediction level increases (from I to P then B), the input data fed for the DCT process become less and less correlated. On the other hand, the DCT Kernel has strong decorrelation property compared to the H.264 Kernel. What we can derive from the "common understanding" is that for input data with strong correlation (I-Frame data) the IK(5,7,3) is preferred, while for input data with less correlation (B-Frame data) the IK(1,2,1) is preferred.

IV. Kernel Selection Process Using a Graphical Approach

The "common understanding" may be too general to properly explain the above observations. Firstly, by examining the residuals that needed to be transform coded (decided by the H.264 encoder – Joint Model (JM) Ver 12.2 [10]), we found that the residuals from different types of frames are not so different in terms of correlation. Having this in mind, in order to explain why only one of the two kernels is overwhelmingly selected, we must investigate the kernel selection process – Rate-Distortion Optimization (RDO) which is shown as follow

$$J = D + \lambda \cdot R \qquad (2)$$

where J is the "decisive cost", D denotes the "quality", R denotes the "cost" and scalar λ is the Lagrange Multiplier. As we can see from (2), a large λ means a more emphasis on the issue of "cost", while a small λ means a more emphasis on the issue of "quality".

The kernel selection process using RDO can be explained by an example as shown in Fig. 1. P_1, P_2 and P_3 are three possible operation points that can be selected. When the criterion of selection is determined (which means that λ is a fixed value), we draw a family of parallel lines passing through all the operation points. The operation point by which the line with the smallest intersection J on D-axis passed is exactly the best one we are searching for. Hence, in Fig. 1, P_1 is the best operation point since the line passing through it has the smallest intersection value as shown by the shaded J_1.

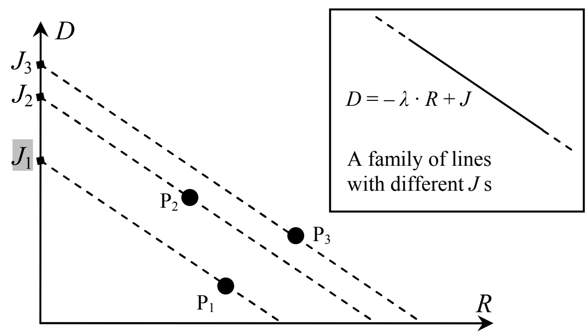

Fig. 1. Visualizing the Kernel Selection Process

By employing the AKM for the H.264, each macroblock will be coded using IK(1,2,1) and IK(5,7,3), and then selected by the RDO. The operation points for a macroblock by IK(1,2,1) and IK(5,7,3) are usually in different locations on the RD-plane. In order to avoid confusion when more than one macroblocks are simultaneously shown on the same RD-plane, we deliberately connect operation points of the same macroblock by a line segment which is often used to represent a macroblock coded by two kernels. As will be shown later, for two kernels with not so large performance gap, the slope of line segment is usually a negative value. For convenience, we denote the negation of slope of line segment by k. For different kernel selection criteria, the relationship between λ and k differs hence the optimal operation point differs (the shaded J represents the optimal cost), which is shown in Fig. 2 (a) and Fig. 2 (b). More specifically, the relationship between λ and k and the subsequent selection of optimal point is concluded in TABLE II.

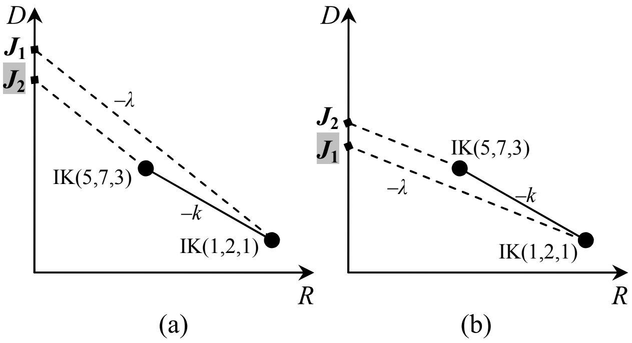

Fig. 2. Selection of the optimal operation point: (a) when $\lambda > k$, the optimal operation point is IK(5,7,3); and (b) when $\lambda < k$, the optimal operation point is IK(1,2,1).

TABLE II
CRITERION OF OPTIMAL OPERATION POINT SELECTION

Relationship	Smallest Intersection	Optimal Operation Point
$\lambda > k$	J_2	Upper-left Point – IK(5,7,3)
$\lambda < k$	J_1	Lower-right Point – IK(1,2,1)

V. A FEATURE EXTRACTED FROM A PAIR OF KERNELS

In the H.264, the RDO for kernel selection is employed in macroblock level, and the MSE of a macroblock is the "quality" and the number of bits of the macroblock is the "cost". The term λ in H.264 is defined as

$$\lambda = s \cdot 2^{(QP-12)/3} \tag{3}$$

where s is called Lambda Weight (LW) and QP is the quantization parameter.

For the H.264, the prediction structure mainly contains three levels – I (no prediction), P (1st order prediction) and B (2nd order prediction). According to [11], in order to get an encoded video sequence with higher coding efficiency, under the same QP, the I-Frame should focus more on quality, while the B-Frame should focus more on using less number of bits. The actual implementation is in accordance with the above idea in the sense that the value of s usually differs from one frame type to another. A common selection may be $s = s_{\text{I-Frame}} = 0.65$, $s = s_{\text{P-Frame}} = 0.68$ and $s = s_{\text{B-Frame}} = 2.00$, which shows the trend of a more emphasis on bitrate saving as the prediction level increases. Even if macroblocks of all types of frames have exactly the same contents, different λ's may select different kernel for each type of frames.

Let us visualize the performances of kernels of all macroblocks for I-, P- and B-Frames in Fig. 3 (a), (c) and (e), respectively. In Fig. 3 (a), each line segment represents a pair of operation points of a macroblock. In each line segment, the top-left one is the operation point of the IK(5,7,3), and the bottom-right one is the operation point of the IK(1,2,1). Since the contents of macroblocks differ from one to another, line segments may span across the whole RD-plane. As we can see from the graph, the line segments are almost pointing to the same direction.

Fig. 3 (b) shows the distribution of the slopes which concentrates at 31.61. Furthermore, 96.95% of k are larger than $\lambda_{\text{I-Frame}}$, which implies that 96.95% macroblocks better be coded with the IK(1,2,1). The interpretation is also similar and valid for P- and B-Frame, except that most macroblocks of the B-Frames are better coded with the IK(5,7,3).

According to our testing, as QP increases (resulted in an increase in λ), k also increases, which continues to preserve the kernel selection tendencies. Using a curve fitting algorithm for different video sequences and QPs, we have found the relationship between k and QP shown as follow

$$k = k_0 \cdot 2^{QP/3} \tag{4}$$

where k_0 is a scalar.

Subtracting (4) from (3), we get

$$\lambda - k = 2^{QP/3} \cdot \left(2^{-4} s - k_0\right) \tag{5}$$

We thus arrive at a crucial conclusion that whether $\lambda > k$ or $\lambda < k$ is independent of the value of QP. Hence we propose to adopt k_0 as the feature which can be extracted from a pair of kernels.

According to various experiments we have carried out (see the Appendix for the other two experimental results which do not show good kernel selection tendencies), we found that this feature does not exist when any of the two kernels is not DCT-like. In other words, the feature only exists provided that both kernels are DCT-like kernels.

A DCT-like kernel is a kernel that is similar to the DCT Kernel as in (1), in a further sense that it preserves the ratios among coefficients as much as possible. The similarity can be indirectly measured by the *Percentage Error* of a/b and c/b of the new kernel to those of the DCT Kernel. After some mathematical derivation, we reached

$$Percentage\_Error = \left| \frac{\sqrt{\frac{1}{2}\left[\left(\frac{c}{b}\right)^2 + 1\right]} + \frac{c}{b}}{\sqrt{\frac{1}{2}\left[\left(\frac{c_{\text{DCT}}}{b_{\text{DCT}}}\right)^2 + 1\right]} + \frac{c_{\text{DCT}}}{b_{\text{DCT}}}} - 1 \right| \tag{6}$$

where b and c are the coefficients mentioned in (1), and b_{DCT} and c_{DCT} are the corresponding coefficients of the DCT Kernel.

The feature is important because 1) it widely exists in pairs of two kernels provided they are DCT-like; and 2) it is independent of QP. It can serve as a theoretical support for us to replace the AKM [8], and turn using only one appropriate kernel for each type of frames which has a unique criterion described by the value of λ. For example, we have two candidate kernels IK(1,2,1) and IK(5,7,3) and we want to determine which kernel is a better choice for each type of frames. The approach for kernel determination is described as follow:

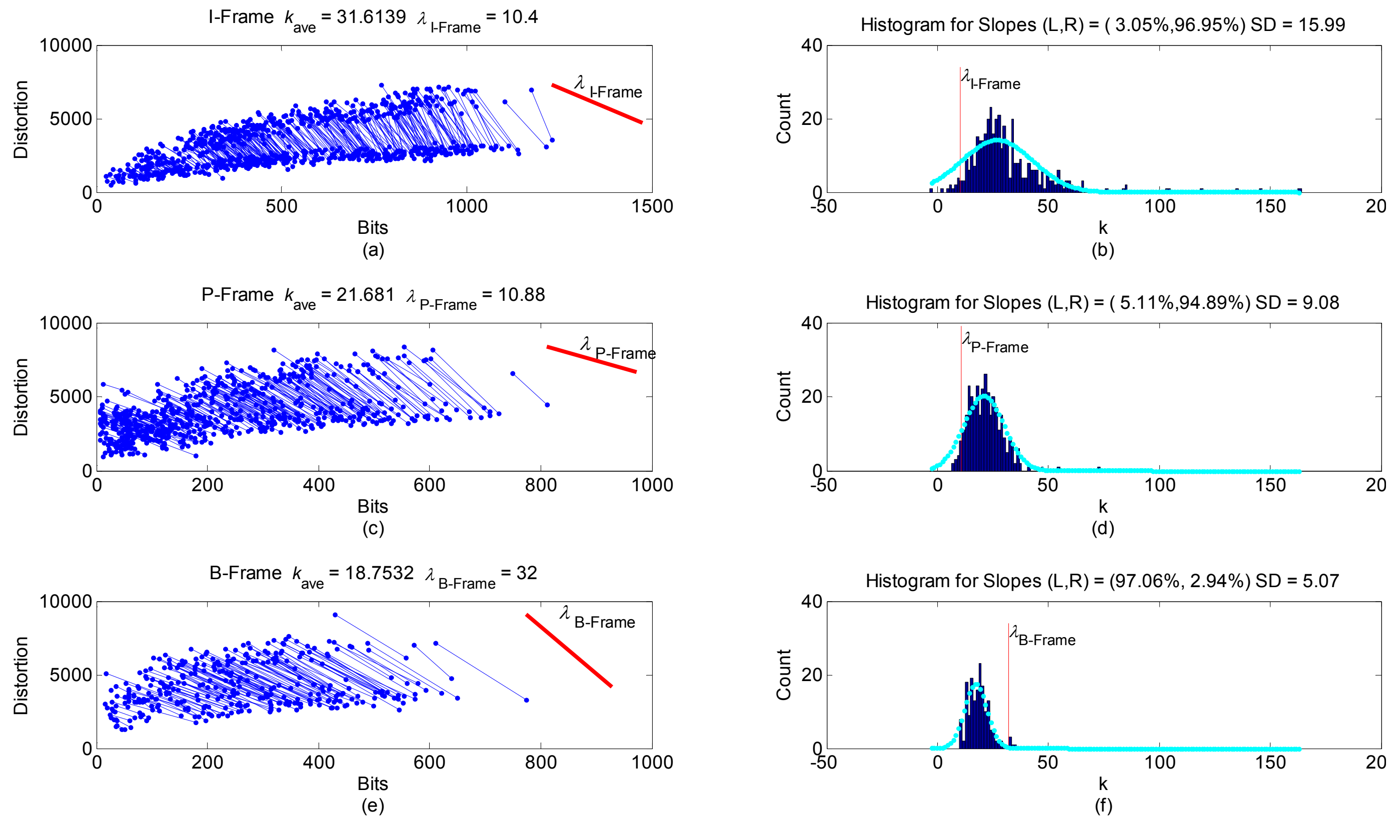

Fig. 3. Visualizing the performances of kernels – IK(1,2,1) and IK(5,7,3) for Sequence *Tempete* at QP = 24. Note that the λ's for kernel selections are shown using red color. (a) Line segments of all macroblock of I-Frame. (b) The distribution of the slopes of line segments of I-Frame. (c) Line segments of all macroblock of P-Frame. (d) The distribution of the slopes of line segments of P-Frame. (e) Line segments of all macroblock of B-Frame. (f) The distribution of the slopes of line segments of B-Frame.

Step1. Select a particular type of frames for kernel determination, for example, I-Frame.

Step2. Set the QP to a reasonable value.

Step3. Encode each macroblock repeatedly using IK(1,2,1) and IK(5,7,3) and record down the MSE and the number of bits used for plotting the corresponding line segment.

Step4. Plot the line segment for every macroblock in an RD-plane, and examine the distribution of slopes of line segments. If the distribution is in single-modal, calculate the average slope of this macroblock; if not, this pair of kernels is not suitable for this algorithm.

Step5. Compare the value of λ with the k_{ave} (average slope), and the preferred kernel is thus determined.

Step6. Return to Step 1 to determine the preferred kernel for other types of frames (i.e. P- or B-Frame) if necessary.

Note that the above steps need only to undergo one particular QP since the selection is mainly independent of QP.

VI. Propose Kernels for the H.264

When we turn to look at the special case – the H.264, we can have a clear conclusion on its kernel determination. Because most kernels of the AKM are DCT-like kernels, one pair of kernels often has a feature which leads the encoder to select one particular kernel for most macroblocks for that type of frames as indicated in Fig. 3. Given that for most macroblocks the best kernel is predetermined by its frame type, it is reasonable to propose using only one particular kernel for all macroblocks of each type of frames since the computational complexity is much lowered and overhead is removed by using a single kernel when the coding efficiency can mainly be maintained.

Hence, we propose using IK(1,2,1) for I- and P-Frames, and IK(5,7,3) for B-Frames. We also evaluate the performance of AKM using IK(5,7,3) as a comparison, since this arrangement has a better performance than the AKM with an alternative sine transform kernel. We implemented our proposed method and the AKM in JM Ver 12.2, and our evaluation is in-line with the "Recommended Simulation Common Conditions" suggested by VCEG [12]. We used the Bjontegaard Metric [13] to measure both changes of Bitrate and PSNR. The evaluation results are shown in the TABLE III.

As we can see from the table, the AKM already outperforms the H.264 Default Scheme, and our proposed scheme performs even better. The proposed method has an average PSNR increase of 0.1526 dB, while the AKM has an average PSNR increase of 0.1491 dB. The bitrate saving for the proposed method can achieve 3.39% on average,

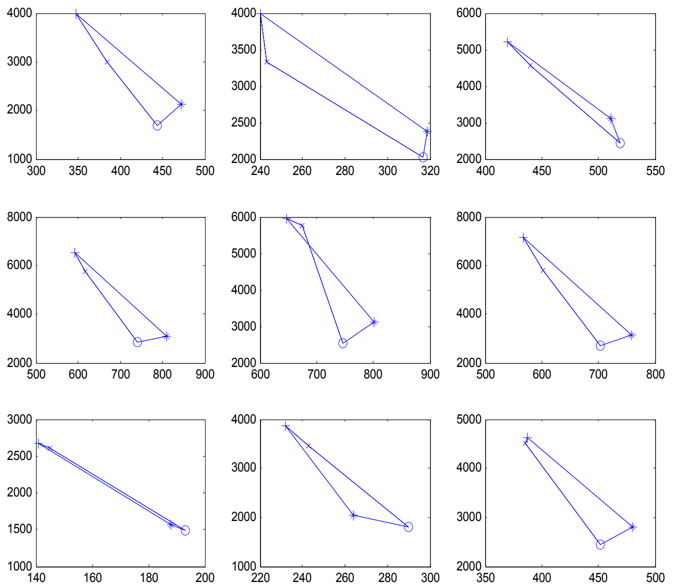

Fig. 5. Nine MLKGs (out of a total of 393 MLKGs) of the I-Frame of Sequence *Tempete* at *QP* = 24.

comparing to 3.36% for the AKM. Note that the employment of AKM also introduces overhead – the signaling bits (although it take only about 0.1% ~ 0.5% of the overall bits of encoded video sequence), hence the improvement of coding efficiency is not so much as shown in the table. So we can conclude that the increase of coding efficiency by employing the proposed method is slightly higher than the increase of coding efficiency by employing the AKM, however, the computational complexity is greatly lowered by the proposed method.

TABLE III
COMPARISON BETWEEN PROPOSED METHOD AND AKM

Sequence		Improvement over the H.264 Default Scheme			
Size	Name	Proposed (Single Kernel)		AKM{(1,2,1)&(5,7,3)}	
		ΔBitrate (%)	ΔPSNR (dB)	ΔBitrate (%)	ΔPSNR (dB)
QCIF	Container	-0.75	0.0346	-1.43	0.0654
	Foreman	-4.28	0.2038	-4.15	0.1980
	Silent	-5.74	0.3034	-5.46	0.2865
CIF	Paris	-6.72	0.3730	-5.82	0.3211
	Foreman	-3.49	0.1487	-3.13	0.1327
	Mobile	-6.20	0.2891	-6.36	0.2970
	Tempete	-5.93	0.2441	-5.75	0.2360
HD (720p)	BigShips	-0.13	0.0031	-0.35	0.0092
	City	-1.46	0.0465	-1.65	0.0539
	Crew	-2.32	0.0594	-3.77	0.0971
	Night	-3.10	0.1101	-2.80	0.1004
	ShuttleStart	-0.61	0.0149	0.35	-0.0087
Average		**-3.39**	**0.1526**	**-3.36**	**0.1491**

VII. USE MACROBLOCK-LEVEL KERNEL GALAXY FOR KERNEL INVESTIGATION FOR FUTURE STANDARDS

As we have mentioned in the above section when λ is given, k_0 is a good feature to let us determine the preferred kernel when a pair of kernels is given. Given that the relationship is robust, it is also intuitive to further expand the pair of two kernels into a group of many kernels in the hope that more kernels can also form a stable pattern. Note that we mainly use DCT-like kernels because the performances of other kernels are not sufficiently good and not so robust as the DCT-like kernels as we have mentioned before. We name the collection of operation points of kernels as a Macroblock-Level Kernel Galaxy (MLKG). As its name indicates, each galaxy shows the capabilities of different kernels encoding the same macroblock. We have carried out tests using the AKM with four DCT-like kernels. We extracted the RD-data for all macroblocks and visualized a portion of them by the MLKGs as shown in Fig. 5. Let us also show the average performances of the four kernels in Fig. 4.

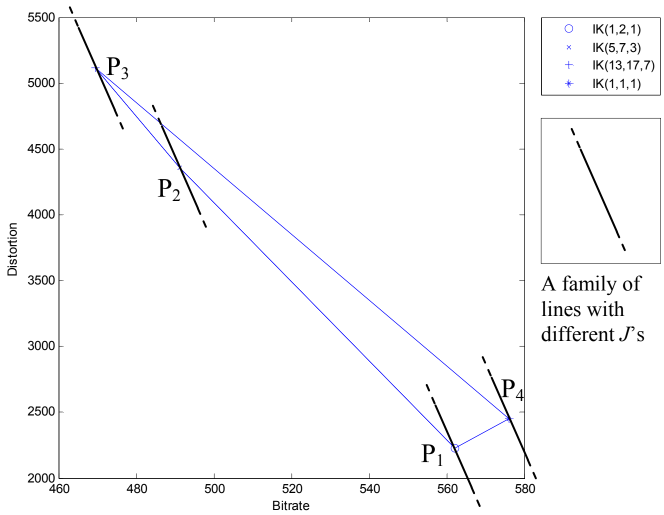

Fig. 4. A Simple Application of the MLKG which shows the various performances of different kernels for the I-Frame of Sequence *Tempete* at *QP* = 24. Each point represents the average of all operation points of one particular kernel. The family of lines represents the kernel selection criterion λ.

According to the experiments (one is shown in Fig. 4), the patterns of MLKGs are mainly stable and similar for all macroblocks, which indicate the feature between two kernels can be generalized to a feature extracted from a group of kernels. This feature is the topology among all operation points. Fig. 4 shows a simple application using MLKG. We aim at choosing a good kernel for efficient coding but at the same time, keep the computational complexity at a reasonable level. The optimal point therefore does not sit at the boundaries of the galaxy (i.e. P_3) but inside it, balancing the coding efficiency and computational complexity. Hence P_2 is the best operation point and subsequently K2 is the preferred kernel for this type of frames.

The experimental results give us a strong confidence in using the MLKG approach to help design single-kernel transform coding process for future video coding standards. We propose the kernel determination process as follow:

Step1. Use one DCT-like kernel to get an optimized λ.

Step2. Use MLKG to selection one kernel from many DCT-like kernels.

Step3. Repeat Step 1 and 2 for different *QP*s to ensure the correctness.

The first step is usually performed for a one kernel standard, to find a λ to optimize the coding efficiency. It is the basic starting point, and our kernel determination process will work around the starting point to do the refinement. We do not expect that the refinement can double the coding efficiency which is impossible, however, it really can achieve even higher performance through the selection of the best kernel.

The MLKG can even visualize the performances of non-DCT-like kernels. However, the relative positions of operation points of these kernels do not form a regular pattern which means that their performances are non-stable.

VIII. Conclusions

In this paper, we make use of the AKM as a start to exploit the kernel selection tendencies. We have found that the slope of a pair of DCT-like kernels forms an important feature for kernel selection. (The in-depth physics behind the feature will be a good subject to be explored.) According to our findings, we also propose using the IK(1,2,1) for I- and P-Frames and the IK(5,7,3) for B-Frames for the H.264. We also generalize the feature extracted from a pair of kernels to a feature extracted from a group of many kernels, and name the collection of corresponding operation points as MLKG. Regarding to the robustness and ease of visualizing the performance, we also propose using MLKG as a guide to design single-kernel transform coding process for future video coding standards, such as the H.265.

Acknowledgment

This work is supported by the Centre for Signal Processing, the Hong Kong Polytechnic University and the Research Grant Council of the Hong Kong SAR Government (PolyU 5267/07E).

Appendix

The RD-graph of kernels IK(1,2,1) and IK(1,1,1) is shown in Fig. 6, and the RD-graph of kernels IK(1,2,1) and IST is shown in Fig. 7. The line segments of either case are pointing to every direction, which means that there is no feature can be extracted from the slopes of the line segments. Note that comparing to the corresponding subfigures in Fig. 7, Fig. 6 (b) and (f) do depict some kernel tendencies. An explanation may be given that the kernel IK(1,1,1) is not fully DCT-like but it is actually more similar to the DCT kernel than the IST Kernel.

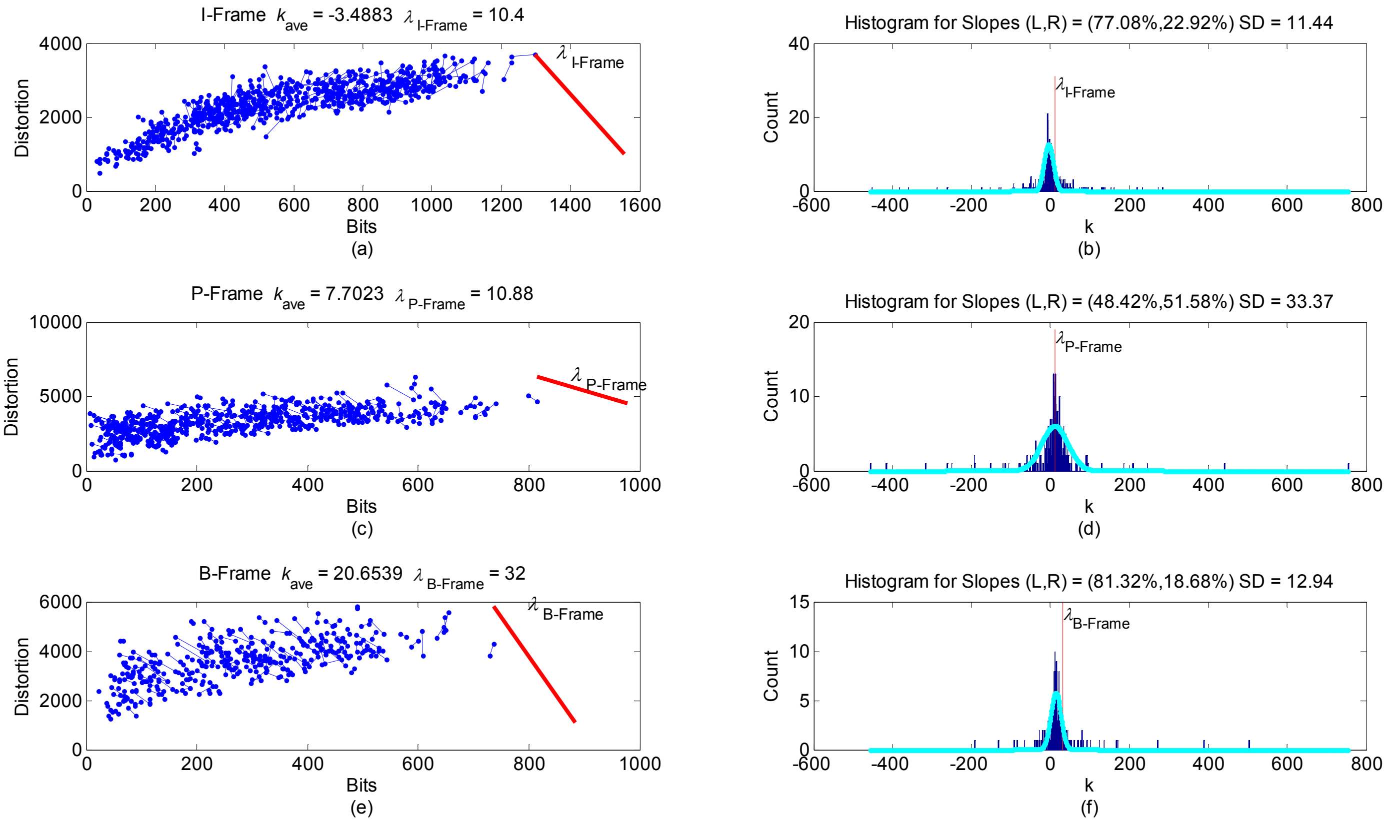

Fig. 6. Visualizing the performances of a DCT-like kernel IK(1,2,1) and a non-DCT-like kernel IK(1,1,1) for Sequence *Tempete* at QP = 24. Note that the λ's for kernel selections are shown using red color. (a) Line segments of all macroblock of I-Frame. (b) The distribution of the slopes of line segments of I-Frame. (c) Line segments of all macroblock of P-Frame. (d) The distribution of the slopes of line segments of P-Frame. (e) Line segments of all macroblock of B-Frame. (f) The distribution of the slopes of line segments of B-Frame.

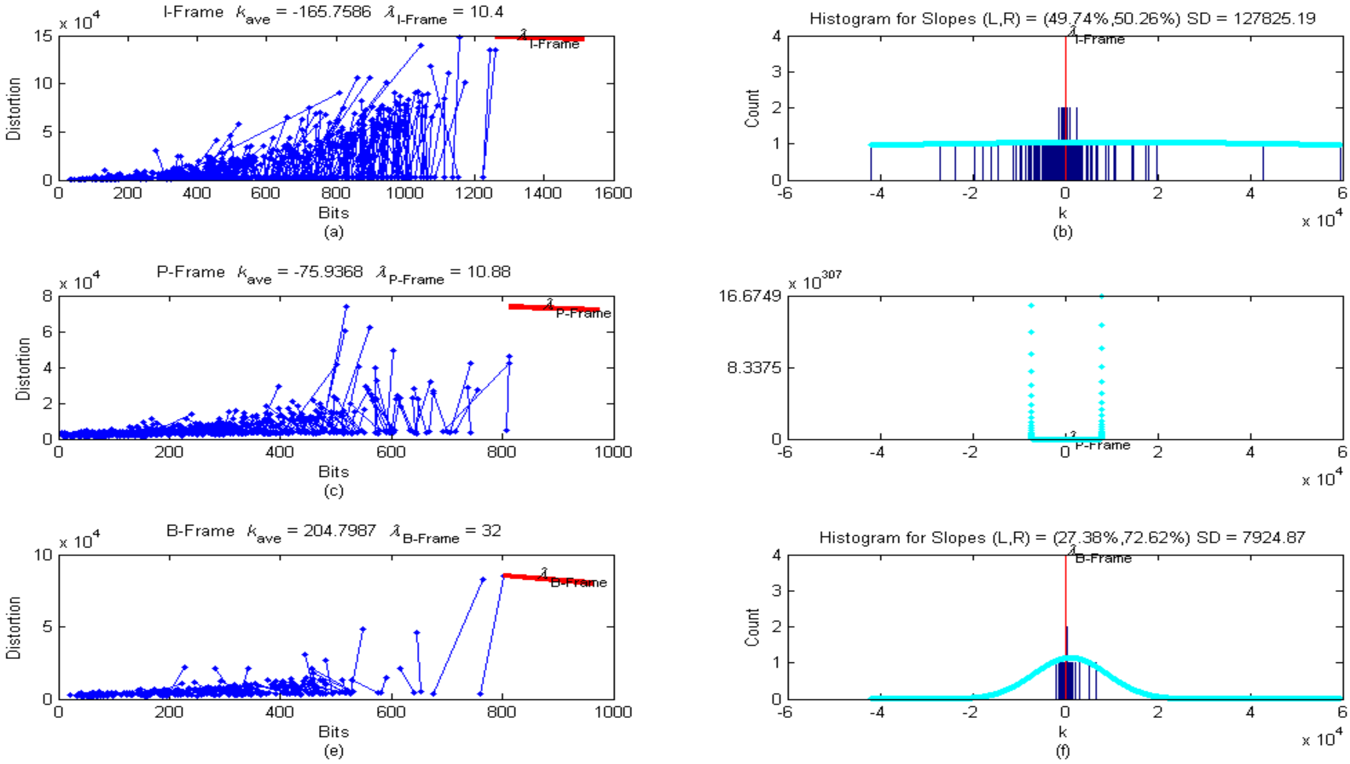

Fig. 7. Visualizing the performances of a DCT-like kernel IK(1,2,1) and a IST Kernel for Sequence *Tempete* at *QP* = 24. Note that the λ's for kernel selections are shown using red color. (a) Line segments of all macroblock of I-Frame. (b) The distribution of the slopes of line segments of I-Frame. (c) Line segments of all macroblock of P-Frame. (d) The distribution of the slopes of line segments of P-Frame. (e) Line segments of all macroblock of B-Frame. (f) The distribution of the slopes of line segments of B-Frame.

REFERENCES

[1] V. Britanak, P. C. Yip and K. R. Rao, Discrete Cosine and Sine Transforms: General properties, Fast algorithms and Integer Approximations, Academic Press, 2006.

[2] Y. L. Chan and W. C. Siu, "Variable Temporal Length 3D Discrete Cosine Transform Coding," IEEE Transactions on Image Processing, vol.6, no.5, pp.758-763, May 1997.

[3] Y. H. Chan and W. C. Siu, "Mixed-Radix Cosine Transform," IEEE Transactions on Signal Processing, vol.41, no.11, pp.3157-61, Nov 1993.

[4] W. K. Cham, "Development of Integer Cosine Transforms by the Principle of Dyadic Symmetry," Communications, Speech and Vision, IEE Proceedings I, vol.136, no.4, pp. 276-282, Aug 1989.

[5] ITU-T Recommendation H.264 and ISO/IEC 14496-10, "AVC for Generic Audiovisual Services," May 2003.

[6] S. C. Lim, D. Y. Kim and Y. L. Lee, "Alternative Transform Based on the Correlation of the Residual Signal," Congress on Image and Signal Processing 2008, vol.1, pp.389-394, 27-30 May 2008.

[7] S. C. Lim, H. Choi, S. Jeong and J. S. Choi, "Integer Sine Transform for Inter Frame," ITU-T SG16, VCEG-AJ12, Oct 2008.

[8] C. W. Wong and W. C. Siu, "Transform Kernel Selection Strategy for the H.264 and Future Video Coding Standards", Submitted to the IEEE Transactions on Circuits and Systems for Video Technology.

[9] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.7, pp. 688-703, July 2003.

[10] H.264/AVC Reference Software Version 12.2. http://iphome.hhi.de/suehring/tml/.

[11] M. Flierl and B. Girod, "Generalized B Pictures and the Draft H.264/AVC Video-Compression Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.7, pp. 587-597, July 2003.

[12] T. K. Tan, G. Sullivan and T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 1," ITU-T SG16, VCEG-AE010, 13-19 Jan 2007.

[13] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16, Doc. VCEG-M33, 26 Mar 2001.