Towards the Next Video Standard: High Efficiency Video Coding

Hsueh-Ming Hang[1], Wen-Hsiao Peng[2], Chia-Hsin Chan[3] and Chun-Chi Chen[4]
[1]Department of Electronics Engineering, National Chiao Tung University
E-mail:hmhang@mail.nctu.edu.tw
[2−4]Department of Computer Science, National Chiao Tung University
E-mail:[2]pawn@mail.si2lab.org, [3]terry0201.cs98g@nctu.edu.tw, [4]cheerchen.cs98g@g2.nctu.edu.tw

***Abstract*—After the profound success of defining H.264/AVC video coding standard in 2002, ITU-T Video Coding Experts Group (VCEG) started a Next-generation Video Coding (NGVC) project. The original target is to achieve 50% bit rate reduction at about the same video quality. In the past a few years, researchers have been very actively searching for new or improved technologies that can achieve this goal. After several years' struggle, in January 2010, the ISO/IEC Motion Picture Expert Group (MPEG) and VCEG jointly issued a call-for-proposal for the "High Efficiency Video Coding (HEVC)". At the April VCEG/MPEG meeting, 27 proposals were evaluated and the results seem to be promising. Consequently, a "new" video standard may be defined in two years. We will present a limited and maybe biased view on this subject.**

I. Introduction

The success of recent multimedia systems such as digital TV and digital camera are often contributed to the standardization of video/audio compression algorithms and the wide spread of personal computer, Internets, and wireless technologies. The advance of digital video compression in the last three decades has produced fruitful results in the past 10 years. Several international image and video coding scenarios have been standardized, for example, ITU H.261/H.263 for video telephony, ISO/IEC JPEG for still images, and ISO/IEC MPEG-1 and MPEG-2 for video CD and digital TV [1][2][3]. After the object- oriented video coding standard, MPEG-4 part 2, was produced, the most significant addition to the video coding standards was H.264/MPEG-4 Advanced Video Coding (AVC) standard finalized in 2003 [3][4]. Since then, many people have tried to design a video coding algorithm more efficient than AVC. In February 2010, 27 proposals submitted to the ITU/ISO joint committee competing for the next generation video standard. The proposal evaluation results in the April standard meeting indicated that a better coding scheme is possible and thus the High Efficiency Video Coding (HEVC) work item was launched.

In the rest of this paper, we will first cover the basic image/video techniques adopted by the international ITU/ISO standards before 2010 in Section II. Then, we will describe the progress of the HEVC standard activities in Section III. In Section IV, we show some comparisons with respect to the coding efficiency and complexity between the currently released HEVC software and the AVC JM software. The main body of this article, Section V, is dedicated to a summary of new tools that may be included in HEVC. At the end, we add a few words on the projection of this standard and the video coding research trends.

II. From JPEG to AVC

Historically, the modern image/video coding standard activity started in 1984 and the target was for video telephony. The output of this activity is CCITT (ITU) H.261. However, in logical order, the simple still image standard (JPEG) will be first described below and then followed by the more complex moving image standards.

The International Standards Organization (ISO) Joint Photographic Experts Troup (JPEG) spent several years in developing an algorithm for compressing still images. In 1987, 10 proposals were evaluated and the adaptive DCT scheme stood out. Although major technical points were agreed in about 1989, the JPEG standard (ISO 10918-1) was formally finalized in 1992 [1][2]. The core of this algorithm is the so-call *transform coding technique*, which converts the digitized picture samples (pixels or pels) into transform coefficients. The correlation of highly redundant pixels in spatial domain is largely removed by DCT. In addition, for most natural pictures, DCT packs the originally scattered power in the spatial domain into a few low frequency components in the transform domain. Transform coefficients are then quantized to reduce the transmission bit rate.

To further reduce the average transmission bit rate, the frequently occurred events (quantized coefficient patterns) are assigned short codes, and the seldom occurred events, long codes. This procedure is the so-called Variable Word-Length Coding (VLC), a modified version of the well-known Huffman code. For typical natural (color) pictures, JPEG algorithm offers a compression ratio of 10 to 20 (or 1 to 2 bits per pixel) with good image quality.

In 1984, CCITT (International Telegraph and Telephone Consultative Committee) started a standard for sending videophone (and videoconference) pictures through ISDN. A set of such standards was finalized in 1990, also known as the *p*x64k standards [1][2]. In addition to the DCT coding technique, the block motion compensation technique is adopted by CCITT

Proceedings of the Second APSIPA Annual Summit and Conference, pages 609–618,
Biopolis, Singapore, 14-17 December 2010.

10-0106090618©2010 APSIPA. All rights reserved.

INTEL-1127
10,142,659

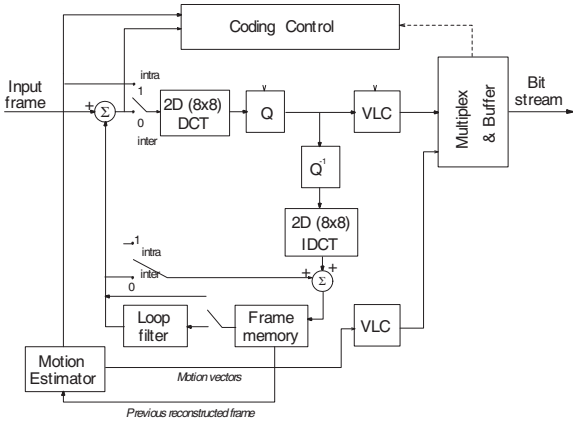

Fig. 1. The H.261 RM8encoder structure.

H.261 [1][2][3]. Fig. 1 shows a typical H.261 encoder (Reference Model 8) [5]. In fact, the same basic coding structure is inherited by all the video standards mentioned in Section I. More precisely, the standards specify only the decoder; however, reference encoders are provided in the standard documents and they are pretty much viewed as the typical encoders in implementation.

To save transmission bandwidth, only the parts of a picture that change from frame to frame are sent. The motion estimation technique calculates the displacement vector of a Macroblock (16 pels by 16 lines) that moves from the previous frame to the current frame. Then, the *prediction errors*, differences between the current frame pixels and the displaced (or motion-compensated) previous frame pixels, are coded and transmitted along with the corresponding displacement vector. In H.261, either the original image block or the prediction error block is DCT-transformed (DCT), quantized (Q), and then VLC-coded.

To further increase the compression efficiency, the ISO Moving Picture Experts Group (MPEG) adopted the rather complicated *motion-compensated interpolation* technique [1][2][3]. This is the main difference between the H.261 algorithm and the MPEG algorithm. Either the previous frame or the future frame (in camera acquisition) or both of them can be used in MPEG to produce the prediction (interpolation) errors. Thus, the so-called *future frame* has to be coded and transmitted before the current frame. Therefore, the transmission order of pictures is different from the order taken at camera. At about 4 Mb/s, MPEG-2 can produce very good quality pictures at regular TV picture resolutions. It thus became the video standards of DVD and digital TV. The ISO MPEG group was established in 1988 and the MPEG-1 video (ISO 11172 part 2) and MPEG-2 video (ISO 13818 part 2) were finalized in 1992 and 1994, respectively.

In 1993, the ITU-T Video Coding Experts Group (VCEG) started new work items. A Near-Term project was targeting at improving H.261 and a Long-Term project was developing a more efficient coding scheme that may be different from H.261. The Near-Term project produced H.263 in 1995 and the Long-Term project (H.26L) led to the well-known H.264/MPEG-4 AVC standard [3][4]. The core technology adopted by AVC is the old hybrid transform coding structure shown in Fig. 1. However, every function block in Fig. 1 was fine-tuned to produce significantly better overall coding results. It has been reported by several studies that for typical TV pictures, AVC reduces the bit rate of MPEG-2 by about 50% at the same visual quality [6][7]. Continuing along the smaller 8×8 block motion compensation trend in MPEG-2 and H.263, AVC allows 16×8 (8×16), 8×4 (4×8), and 4×4 partitions for motion prediction [4].

III. Joint Collaborative Team Activities

A call-for-proposal was issued in 1998 by ITU VCEG. The goal is a low-bit rate, low delay video codec, which was called H.26L then. After a few years of development, VCEG and the MPEG video group formed the Joint Video Team (JVT) in 2001. The final AVC standard was produced by JVT in 2003. Since then, VCEG launched an exploration activity on the Next-Generation Video Coding (NGVC) project. Its target was to further reduce the video compression bit rate by 50% over AVC. However, the high compression efficiency of AVC is hard to beat. Many new algorithms were proposed to further improve the coding efficiency, but few could show significantly better performance. As time goes by, the AVC tools were further refined and people noticed some modifications were able to produce a certain amount of improvement. These modifications were collected and formed a piece of software called *Key Technical Areas* (KTA) [8] since 2005.

The KTA scheme is more or less an expansion of AVC but with careful tuning on all the components. In June 2009, the MPEG video group held a call-for-evidence activity. Several schemes based on KTA were submitted. It showed that a 30% or so coding gain over AVC was possible on higher resolution videos. Thus, ITU VCEG and MPEG worked together again and formed the so-called *Joint Collaborative Team on Video Coding (JCT-VC)* in January 2010. A joint Call-for-Proposal (CfP) for High Performance Video Coding (HVC) was issued. All the proponents had to submit their test material before February 22, 2010.

The main goals of HVC stated in the requirement document are (a) coding performance on high resolution pictures, (b) picture size up to 8K×4K, (c) low delay, and (d) low complexity [9]. Although the MPEG requirement document does not specify the compression efficiency improvement, the ITU NGVC requirements do hope that there is a 50% bit rate reduction over AVC [10].

The joint CfP for HVC is a quite lengthy document [11]. There are 5 Classes of test sequences: (A) 2560×1600 cropped from 4K×2K, 2 sequences; (B) 1920×1080p, 24/50-60 fps, 5 sequences; (C) 832×480 WVGA, 4 sequences; (D) 416×240 WQVGA, 4 sequences; and (E) 1280×720p, 50-60 fps, 3 sequences. A number of test points (conditions) are specified. They belong to two constraint categories. Constraint 1 (CS1) is the *Random Access* setting for Classes A to D; a delay of 8-picture GOP is allowed. Constraint 2 (CS2) is the *Low*

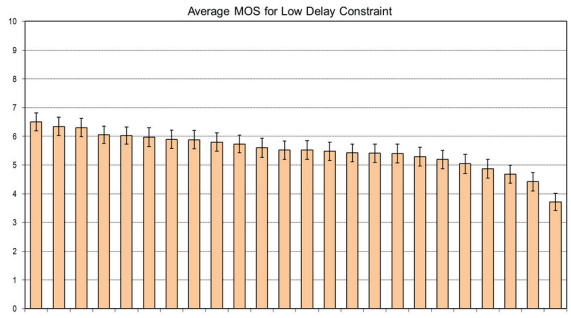

Fig. 2. Overall average MOS results over all Classes for CS2.

Delay setting for Classes B to E, and no picture re-ordering is allowed. For comparison purpose, three *Anchors* are defined. They are the same AVC coding schemes with different coding profiles and parameters. The Alpha Anchor meets CS1 conditions, and the Beta and Gamma Anchors meet CS2. Since the coding results of three Anchors were published before the CfP due date, all submissions are at least as good as the Anchors.

In total, 27 proposals, which is historical high in MPEG CfP competitions, were submitted to JCT-VC in February and the subjective image quality evaluation was done in March. The evaluation results were discussed in the April JCT-VC meeting at Dresden, Germany. A number of key players in video coding community participated in this competition. Objectively, the top-performers achieve 40% BD-rate [12] savings on CS1, 40% and 55% on CS2. Also, the detailed description of the subjective evaluation results are given in [13]. Limited by space, we copy only one summary plot that shows the coding performance of all proposals including the Anchors. Fig. 2 is the average Mean Opinion Scores (MOS) of 27 proposals plus the Beta and Gamma Anchors for CS2. The Anchors are the lowest two on the right margin in both figures. It is clear that the best proposal is quite a bit better than the Anchors. Similar observations can be found for CS1 coding conditions. Although there is no single proposal did the best for all pictures at all rate points (5 rate points for each sequence), the *good* schemes together are quite promising. Therefore, it was stated in the conclusion of [13] that "for a considerable number of test points, the subjective quality of the proposal encoding was as good, for the best performing proposals, as the quality of the anchors with roughly double the bit rate". However, when we examine the techniques used by all these proposals, "all proposed algorithms were based on the traditional hybrid coding approach combining motion-compensated prediction between video frames with intra-picture prediction, closed-loop operation with in-loop filtering, 2D transformation of the spatial residual signals, and advanced adaptive entropy coding" [13]. These schemes are roughly the expansion and refinements of KTA, which is an extension of AVC. Because of the encouraging testing results, JCT-VC is constructing a *Test Model under Consideration* (TMuC) [14] described in the next section. If the standardization process runs smoothly, we may have a new video standard in two years.

TABLE I
THE DIFFERENCE OF TOOL SELECTIONS BETWEEN HIGH EFFICIENCY AND LOW COMPLEXITY SETTINGS.

Features	High Efficiency	Low Complexity
Transform partitioning	Quad-tree TU [16]	
Motion Sharing	Merge Mode [16]	
Intra Pre-filtering	Adaptive Intra Smoothing [16]	
Intra Prediction	Combined [17], angular [18], planar [18]	
Directional Transform	MDDT [19], Rotational Transform [20]	
Deblocking Filter [18]	ON	
Adaptive Scanning [19]	ON	
Entropy Coding	PIPE [16]	CAVLC [18]
Interpolation Filter	12-tap SIFO [19]	Directional Filter [18]
Adaptive MV Res. [19]	ON	OFF
Adaptive Loop Filter [19]	ON	OFF
IBDI [21]	4 bits	0 bit

IV. CURRENT TMUC STATUS

A. TMuC

After the CfP competition in the 1st JCT-VC meeting, TMuC is constructed mainly from the best performer's codebase and the other top-performing HEVC proposals. The tools currently included in TMuC may not get into the JCT-VC committee's final *Test Model*[1]. Rather, these tools are merely a preliminary selection and require further evaluation and justification. In the current status, TMuC serves as a good starting point at the very beginning of the collaborative phase, and aims at creating a minimum set of well-tested tools to establish the "*Test Model 1.0*".

JCT-VC committee specifies 6 reference configurations [15] for Tool Experiments (TE). Five test scenarios are classified into two groups: (a) *High Efficiency* (HE) and *Low Complexity* (LC) settings; and (b) *Intra Only*, *Random Access*, and *Low Delay* settings. Six test conditions (or configurations) are formed by picking up one from the first group and one form the second group. Under these testing conditions, the experimental results are expected to provide insights of the contribution of a specific tool to the overall performance of TMuC. It also helps us in defining the profiles for different applications. Table I shows the tool selections in group (a) in TMuC. The HE setting aims at achieving the high coding efficiency close to that of the best performing proposal, while the LC setting intends to lower the complexity close to that of the lowest complexity proposals with a relatively high coding efficiency. Different settings in (b) use different coding structures to achieve the target functions or features. Random Access and Low Delay settings are the same as that in CfP, and the Intra Only scenario contains only the intra-coded frames. Some experimental results using the reference configurations are in Section IV-B and IV-C, to compare TMuC with JM.

B. Compression Performance of TMuC

To see the current TMuC coding performance, experiments are conducted on TMuC 0.7 [14] in comparison with the CfP Anchors generated by JM 16.2 [22]. We exclude *Intra Only* settings and apply the other 4 configurations in [15]. The

[1] *Test Model* is a common test platform such as the JM software for AVC.

TABLE II
SUMMARY OF TMUC BD-RATE SAVINGS.

		Random Access			*Low Delay*		
		vs Alpha		**HE vs**	**vs Gamma**		**HE vs**
Class	**Seq**	**HE**	**LC**	**LC**	**HE**	**LC**	**LC**
Class A	S01	38.9	21.8	21.6	N/A	N/A	N/A
2560×1600	S02	24.9	4.8	19.5	N/A	N/A	N/A
Class B	S03	45.6	29.5	22.9	56.7	45.6	19.7
1920×1080	S04	32.3	14.3	21.1	40.8	25.4	20.5
	S05	40.1	21.0	23.5	48.2	32.6	20.3
	S06	45.3	27.5	23.9	51.2	36.2	23.3
	S07	48.8	19.9	33.2	63.6	38.9	34.6
Class C	S08	39.5	24.0	20.4	44.7	28.2	21.3
832×480	S09	37.4	20.5	21.4	41.5	24.8	21.9
	S10	36.7	13.1	30.0	38.6	17.9	29.2
	S11	34.0	21.9	15.7	35.1	24.1	14.2
Class D	S12	29.3	16.7	15.3	32.6	22.3	13.4
416×240	S13	49.4	3.8	47.2	58.1	10.1	48.3
	S14	30.4	10.6	22.5	32.1	13.3	22.3
	S15	26.4	12.6	15.8	24.8	12.4	14.3
Class E	S16	N/A	N/A	N/A	53.8	26.8	32.6
1280×720	S17	N/A	N/A	N/A	50.9	16.3	36.6
	S18	N/A	N/A	N/A	54.3	28.0	32.2
Total Avg.		*37.2*	*17.5*	*23.6*	*45.4*	*25.2*	*25.3*

results are compared with CfP Alpha and Gamma Anchors. To provide a fair comparison, Gamma Anchor is selected instead of Beta since its coding structure (IPPP) is similar to that of TMuC Low Delay setting (IBBB).

Table II summarizes the BD-rate [12] savings for each test sequence. TMuC achieves 4%∼64% BD-rate savings as compared with JM. Roughly speaking, the coding gain increases with the increasing sequence resolution. However, the coding gain of Class A is lower than that of Class B. The reason may be the immature camera acquisition technology, which results in high capturing noise at ultra-high definition (UHD) resolutions. This is also the reason why Class A did not go under subjective tests in CfP evaluation. Interestingly, some sequences in Class D have significant coding gains, which indicate that the coding tools in TMuC also work well on low resolution sequences.

Comparing the HE and LC settings, the HE setting always outperforms the LC setting for more than 13%. Particularly, for S13, the LC setting has a significant loss in coding gain in comparison with the HE setting. The coding gain drops of the Low Delay case ranges from 58% to 10%. This anomaly may due to disabling some coding tools at the LC setting.

Fig. 3 shows the coded pictures using the CfP Anchors and TMuC. Apparently, in addition to the objective PSNR metric, the subjective image quality has been significantly improved. Since TMuC is in its early stage of development, further improvement is expected in later versions.

C. Encoding/Decoding Complexity of TMuC

Since a formal complexity measuring index for the TMuC software is not established yet, we try to roughly measure the complexity using the encoding and the decoding execution time. Note that the variations in execution platform and compiler optimization may lead to different results.

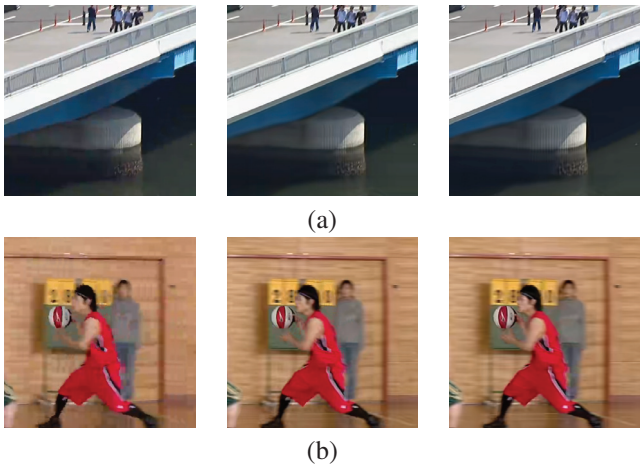

Fig. 3. Subjective comparisons between CfP Anchor (left), TMuC LC (middle), and HE (right) settings at the lowest rate points of (a) the Random Access setting (S07 frame #290), and (b) the Low Delay setting (S06 frame #7).

It is observed that the LC setting indeed shows lower complexity in both encoding and decoding time. The HE setting is, on average, about 3 times slower than the LC setting for encoding, and 1.5 times slower for decoding. As compared with JM in decoding time, TMuC is apparently slower for at least 2.8 times. Especially, for Class D, TMuC is even much slower than JM for more than 11 times; however, TMuC still can run on a typical PC at 10∼15 fps. However, for large-size pictures, the decoder produces only 2∼3 fps for Class A and B. To sum up, TMuC also has plenty of room for improvement in computational complexity.

V. CURRENT HEVC TOOLS SUMMARY

In the conventional video coding standards up to now, most coding tools use fixed parameters or operations to simplify implementation. For example, half-pel interpolation is done by a fixed 6-tap FIR filter in AVC. However, the UHD video contents show strong signal variations and thus the current fixed-parameter coding tools are unable to produce the best possible results. Therefore, the content- and context-adaptive tools emerge in the next-generation video coding design. They change coding parameters on the fly to optimize the coding performance for time and spatial varying signals. Because the adaptive processes usually involve multi-pass encoding optimization, a massively parallel processing architecture becomes increasingly important for real-time implementation.

Based on the above observations, most HEVC proposals use highly adaptive and complex coding tools. Also, the parallel processing features are emphasized. As described earlier, the top-performers outperform the Anchors quite a bit both objectively and subjectively. How could these proposals with the same basic structure as AVC achieve such a high performance? We summarize the new tool features in Table III. Details are given in the following sections.

A. Coding, Prediction and Transform Partitioning

Block-based hybrid video coding structure is the core of all the current video coding standards. Its basic unit for

TABLE III
COMPARISONS OF TOOL FEATURES BETWEEN AVC AND HEVC.

Features	H.264/AVC	HEVC	Section
		Coding, Prediction and Transform Partitioning	
Coding Partitioning	16×16 macroblock	Variable, large size	V-A
Prediction Partitioning	Quadtree-based structure	Irregular, large size	V-A
Transform Partitioning	4×4 and 8×8	Rectangular, large size	V-A
		Motion	
MVp Derivation	Median	MV competition	V-B1
Motion Inference	B_DIRECT, SKIP	P_DIRECT; Enhanced B_DIRECT; SKIP; Template matching	V-B2, V-B3
Motion Sharing	No	Yes	V-B4
		Inter Prediction	
Sub-pel Interpolation Filter	6-tap FIR; bilinear filter	Fixed filter; Weiner-based adaptive filter	V-C1
Parametric OBMC	No	Yes	V-C2
MV Precision	1/4-pel	1/2-, 1/4-, 1/8-, 1/12-pel adaptive	V-C3
Weighted Prediction	Signaled at slice level	Signaled at partition level; Modified offset; Illuminance prediction	V-C4
Spatial-Temporal Prediction	No	Yes. Intra prediction for inter residual	V-C5
		Intra Prediction	
Short-term Prediction	No. Only long-term prediction	Yes. Minimize distance between reference and predicted pixels	V-D1, V-D2
Texture Synthesis	No	Yes. Template matching average	V-D3
Pre-filtering	High-profile Intra 8×8 only	On/Off for all block partitions	V-D4
Post-filtering	No	Yes. Filters applied on predictor	V-D4
Plane Prediction	Yes	Yes. Bilinear; Plane-fitting	V-D5
Chroma Prediction	Independent prediction	Refer to segment information of reconstructed luma samples	V-D6
Directional Prediction	At most 8 directions	More than 8 directions	V-D7
		Transform Coding and Quantization	
Directional Transform	No. Only integer DCT	Yes. MDDT; Switchable transform; Rotational transform	V-E1, V-E2
Quantization Matrix Adaptation	No. Fixed weighting matrix	Yes. Context-adaptive selection of weighting matrices	V-F
		In-loop Filter	
De-blocking Filter	Horizontal and vertical edges	Simplified design; De-banding algorithm	V-G2
Adaptive Loop Filter	No	Yes	V-G1
		Entropy Coding	
Parallelization	No. Only serial processing	Yes. Entropy slice-, syntax-, bin-level parallelization	V-H1
Entropy Coder	VLC; CAVLC; CABAC	Modified CAVLC; CABAC; V2V	V-H2, V-H3
Adaptive Coefficient Scanning	No. Only zig-zag scanning	Yes. Switchable scanning order	V-H4
		Increase Calculation Accuracy	
Bit Depth	8 bits	Internal bit-depth increasing; Dynamic data range extension	V-I

compression, referred as *coding unit* (CU), is usually a fixed N×N pixel square region of a frame and it may contain several *prediction units* (PU) and *transform units* (TU) for inter/intra prediction and transform/residual coding, respectively. In AVC, its CU is a Macroblock (MB), which covers 16×6 luma and 8×8 chroma samples. Its PUs are the various MB partitions having a square or rectangular shape with several sizes. The TUs, on the other hand, always have a square shape, although their sizes can vary too. Particularly, TUs generally have a smaller size than PUs and are always aligned with PUs in order not to across their boundaries. This is because the residual signals of different PUs tend to be uncorrelated.

It is seen in many HEVC proposals that the size and shape of CU, PU and TU as well as their mutual relationship now become more flexible. Inclusion of CUs, PUs, and TUs of larger sizes is a common and critical theme in most proposals. This straightforward extension is to handle the smooth textures in a local area of high-resolution natural videos. Proposals in [17][20] even allow the size of CU to be variable through a tree structure segmentation. Such flexibility, allowing a 16×16 partition to have mixed inter and intra predictions, is not possible in the prior coding standards. To make the representation of objects more accurate and efficient, some proposals [19][20][23][24][25] offer irregular shape PUs (see Fig. 4). Interestingly, there are two extremes in the TU design. While [17][20] provides an option for a TU to across multiple PUs, [16] proposes dividing a PU by quad-tree partition so that TUs are smaller and adjustable in size.

B. Sophisticated Motion Compensation and Parameter Coding

In AVC, the motion-compensated predictor for each MB or sub-MB partition generates motion vectors (MVs) and, possibly, reference frame indices. These motion parameters are transmitted to the decoder and thus they are embedded in

the compressed bit-stream and constitute a significant portion thereof, especially at the low bit-rates. To reduce motion information, advanced MV coding techniques with sophisticated operations thus appear in many proposals.

1) Motion Vector Competition: The MV in AVC is predictive-coded by a motion vector predictor (MVp). Motion vector competition aims at finding a better MVp from an extended MV set to improve coding efficiency. This MV set is composed of previously coded MVs of nearby partitions (blocks) and of temporally co-located partitions. The MVp having the best rate-distortion (R-D) performance is chosen and sent [17][20][24][26][27][28]. To reduce overhead, [23] provides an implicit signaling mechanism, in which the MVp is chosen to achieve the minimal template matching error (see Section V-B3). Note that the candidate MVs may be linearly scaled to account for the varying temporal distance between their respective reference picture and current picture [19][23][29][30][31][32].

Rather than enlarging the MV candidate set, [16] invents an interleaved MV prediction method, which uses the vertical component of a MV, coded in the same manner as AVC, to guide the selection of MVp for its horizontal component.

2) Enhanced SKIP and B_DIRECT Modes: SKIP and B_DIRECT modes are two motion inference methods in AVC. When a MB is coded in either mode, no motion parameters are transmitted. In the case of a skipped MB, the residual samples are also omitted. Both are proven efficient for low bit-rate coding and have been extended or altered in a number of ways in the HEVC proposals. For example, [30] introduces a partial SKIP mode, which applies the notion of SKIP prediction at the partition (or PU) level. In [29], a flag is transmitted for each B_DIRECT MB (or CU) to indicate whether the MVs are inferred with the spatial or the temporal method.[17][20] further incorporates forward and backward uni-directional predictions into the B_DIRECT methods. They also propose a P_DIRECT mode, which permits residual samples to be sent for a skipped P-block. On the other hand, [23][24][25] conduct a candidate set of MV pairs similar to that of the MV competition. One MV pair is then selected to be the MV pair for B_DIRECT mode.

3) Template Matching Prediction (TMP): TMP provides another way of estimating the motion information on the decoder side. The concept of TMP is shown in Fig. 5. It finds the predictor for a target block B by minimizing the prediction error over the pixels in its immediate L-shaped neighborhood, T (usually termed the template). When viewed from motion compensation perspective, it is equivalent to treating the MV found by template matching as the target block MV. Since this operation uses only the reconstructed pixels, the decoder can produce the same predictor as the encoder without sending any motion information. TMP technique can be found in several proposals [23][29][31][33].

4) Motion Information Sharing: Motion information sharing allows a PU to reuse the motion parameters of its neighboring PUs. For example, in [16] a PU uses the motion information of the PU on its top or to its left. Furthermore, the parameters for different partitions in a PU can be deduced from different neighboring PUs as proposed in [27].

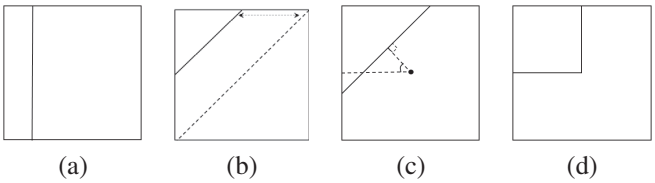

Fig. 4. Irregular PU partition examples of (a) asymmetric [20], (b) flexible [23], (c) geometric [19] and (d) diagonal [24][25] partitionings.

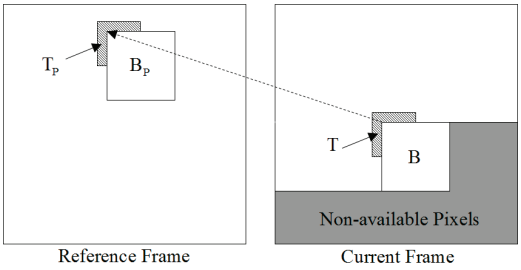

Fig. 5. General concept of TMP.

C. Inter Prediction

Inter prediction is crucial to the overall coding performance. It thus becomes another key focus of improvement. Although no fundamental changes were made to the current prediction concept, there are many variants, which could potentially contribute to new designs. Here we give a brief summary of some noteworthy new techniques.

1) Sub-pel Interpolation: The existing sub-pel interpolation method has been improved by replacing the fixed filters by the adaptive ones or by redesigning the filter coefficients. Several proposals adaptively update interpolation filters by the least squares method in order to minimize the prediction errors of each video frame. In [19][27][28][29][29], multiple sets of filters are transmitted for an adaptive selection at slice or partition level. The extra overheads are reduced by making use of the symmetry properties of these filters.

In addition to adjusting filters on the fly, some redesigned filters are proposed. The schemes in [18][19][28] increase the precision for filtering operations. In [19], not only multiple filters but also a set of derived DC offsets can be selected for each sub-pel position. In [16], filters are derived based on the maximal-order interpolation of minimal support (MOMS). Particularly, [20] provides a filter design framework that can generate filters at any sub-pel position.

2) Parametric OBMC: [33] introduces a parametric overlapped block motion compensation (POBMC) technique to improve inter-frame prediction. It extends the notion of OBMC in H.263 to accommodate the variable block-size motion partitions in AVC. This approach looks for the optimal weights associated with different MVs as functions of the distances between the predicted pixel and its nearby block centers, where these MVs are located. This far-reaching generalization

provides a generic reconstruction framework, allowing the MVs associated with multiple motion partitions of arbitrary shape to be optimally constructed for motion compensation.

3) Adaptive MV Precision: In AVC, the MV resolution is fixed at 1/4-pel precision. Although a higher precision such as 1/8-pel [34] can further reduce prediction error, sometimes the bit-rate increase outweighs the prediction performance due to additional MV coding overheads. Some HEVC proposals signal the MV precisions in order to strike a balance between motion accuracy and MV coding bits. In [16], the MV precision is adaptive at the slice level. Moreover, the MV precision is signaled per MV in [19][20].

4) Weighted Prediction and Illumination Compensation: Weighted prediction performs a linear operation on the predictor, usually with an DC offset, to generate a better prediction result. In [24], the weighted prediction coefficients are switchable for each CU. In [8][21], a new offset is generated by subtracting the average value of the coded picture from that of the reference picture. In [23][30], the luma values of current block are compensated by the difference between the average of neighboring samples of the current block and that of the reference block.

5) Spatial-temporal prediction: In [23], inter prediction residual is compensated through intra prediction. The intra prediction reference is generated by the difference between the current and the reference blocks' neighboring pixels.

D. Intra Prediction

The AVC intra prediction tool provides DC and several directional modes for predicting variable-size blocks. The predictor is linearly generated from target block's neighboring L-shaped coded pixels. However, this prediction scheme has several inherent weaknesses: (a) Poor performance inevitably incurs when the distances between the reference and the predicted pixels increase. (b) The straightforward design of extrapolation filters is incapable of synthesizing periodical and complex textures. (c) Artificial edges, which are not usually seen in nature scenes, appear along the directions of intra prediction. Based on the above investigations, many tools are proposed to alleviate these problems.

1) Line-based Prediction: Since the intra prediction error tends to be larger for farther away pixels, some proposals try to minimize the distance between the reference and target pixels. Line-based predictions, as illustrated in Fig. 6 (a), divide a 16×16 block into 1×16, 16×1, 2×8 or 8×2 partitions rather than the conventional square partitions and sequentially codes each partition to ensure that successive partitions can refer exactly to its neighboring pixels [23][35]. Another proposal, termed the recursive intra prediction [36], is analogous to [23] except that the successive partitions are extrapolated by referring to the predictor of its preceding partition, and only 1×16 and 16×1 partitions are available.

2) Pyramid and Interleaved Prediction: The block-based pyramid prediction [24] firstly encodes a down-sampled version of the current block, which will then be reconstructed and upsampled to serve as the final predictor. The resample-based intra prediction [23] encodes an interleaved block shown in Fig. 6 (b). The A-pels in a block are coded first. Then, the predictors of B- and C-pels are formed by referring to the reconstructed A-pels. After A-, B- and C-pels in a block are coded, the D-pels can be predicted from all of their surrounding pixels.

3) Template Matching Average: The TMP technique can also be applied to the intra frames as show in Fig. 6 (c), aiming at predicting the periodical and the complex textures. To further reduce the estimation error variance, the template matching average (TMA) [27][37] averages the first N candidate blocks that have the lowest template prediction errors, to form the predictor. In the line-based TMA [35] the target block is degenerated into a straight line. Then TMP is performed on each target line whose coding result can be used afterwards in the prediction of successive target lines. In [20], the pixel-based recursive template matching (PTM) is employed recursively for each target pixel in the block in the raster-scan order. In addition to the coded pixels in the search range, the predicted pixels are included in the template as the successive target pixels.

4) Pre- and Post-filtering: The directional patterns of AVC intra prediction extrapolate directional textures by referring to the coded neighboring pixels. However, the synthesized texture may contain artificial edges along the direction perpendicular to the selected prediction direction. The pre- and post-filtering processes are thus introduced respectively for reference samples and predictors to alleviate this problem.

The pre-filtering process, specified in the AVC High Profile, employs a low-pass filter on the reference samples prior to the intra 8×8 prediction. The method in [16] extends this pre-filtering process to all partitions except for the intra 4×4. On the other hand, many post-filter processes are proposed. The initial predictor is filtered by a separable 3×3 Gaussian kernel in [36] and by an average filter applied to the current and neighboring pixels in [20]. The block-based post-filtering is done by a weighted sum of the initial predictor and the neighboring coded blocks in [29]. Yet in [17] the leaky prediction is formed by a weighted sum of the initial predictor and the original block.

5) Plane Prediction: The plane mode prediction, which aims at producing smoothly-varying textures, is also improved. In [18] the bottom-right most pixel is signaled explicitly to linearly interpolate the right-most and bottom pixels. The rest pixels are then interpolated bi-linearly. In [25], firstly a 3D plane surface is derived from fitting the values of the neighboring reconstructed pixels $\mathcal{P}_0$ in Fig. 6 (d). Then the predictor of current block $\mathcal{P}_1$ is created from fitting its pixel values to this surface.

6) Chroma prediction: In general, there should be no correlation between luma and chroma values, whereas this is not true for the textural regions whose segmentation information inferred from the luma component can be used for improving the chroma prediction. A modified DC mode for chroma prediction [20] exploits the segmentation information from

a down-sampled luma block, which is previously coded, to separate the corresponding chroma block into two irregular parts. The luma segmentation map is generated by the thresholding method using the DC mean value. After that, each part is independently estimated by averaging the reference pixels within the same segmented region.

7) Extended Directional Prediction: Because the 8 directional modes defined in AVC may not sufficiently represent all possible directional patterns, various approaches are proposed to increase the number of prediction directions. One is simply increasing the number of directions, which delineate the finer granularity in producing a more precise estimation of directional patterns [17][18][20]. Another proposal, as shown in Fig. 6 (e), is to find a vector with the largest magnitude of the 2-norm of the gradient field constructed from the neighboring reconstructed blocks. The isophotal direction perpendicular to the chosen vector indicates a special directional mode, which shares the same mode number with the DC mode, and will take effect only if its magnitude exceeds a pre-defined threshold [27]. The bi-directional intra prediction (BIP) [21] deduces the predictor from averaging the prediction results of two different modes. Moreover, in anticipation of further improving the predictive coding efficiency, the coding order of BIP is changed in the order as depicted in Fig. 6 (f); hence, additional references at the bottom and/or right sides are available for A, B and C.

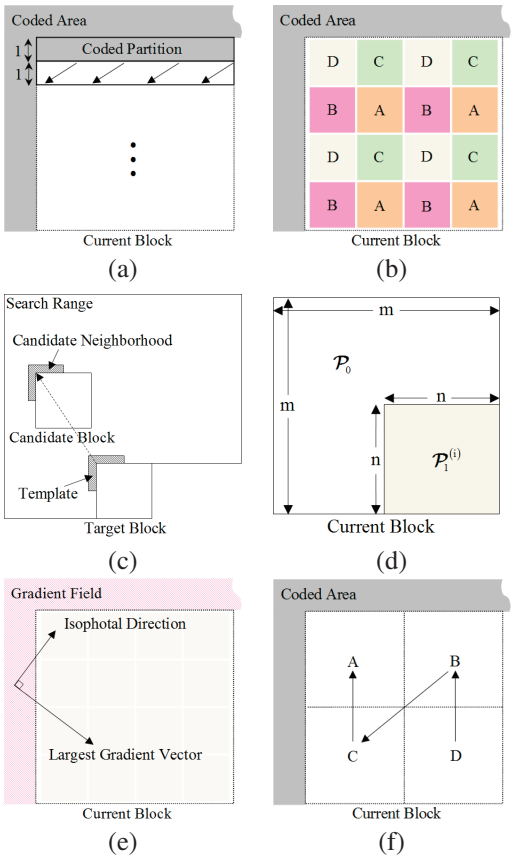

Fig. 6. (a) 16×1 line-based prediction (diagonal down-left mode) [23][35], (b) resample-based intra prediction [23], (c) template matching intra prediction [37], (d) parametric planar prediction and iterative prediction [25], (e) edge-based directional prediction [27] and (f) prediction order of BIP [21].

E. Transform Coding

The transform coding converts inter/intra prediction residuals to the frequency domain in order to decorrelate and compact the residual signals. However, the DCT basis is not optimal for various directional patterns in residual signals. The transform basis should be made adaptable to the statistical variation of realizations. Therefore, anticipation of a need for better transform coding tools leads to redesigning the existing DCT-based coding for further optimizing the energy compaction of residual signals.

1) Mode-dependent Directional Transform: The mode-dependent directional transform (MDDT) [38] is widely used in many HEVC proposals since it has been proven to be effective for decorrelating the redundancies along the directions of intra prediction. In MDDT, each intra prediction mode is coupled with an unique pair of transform matrices, which is derived from the off-line training processes of Karhunen-Loéve transform (KLT), for the strongly mode-dependent residual signals. In order to lower the hardware cost, the orthogonal MDDT [39] forces the column and row transform matrices to be the same in the training processes. This slight change can save half of hardware area or memory usage.

However, even for a given intra prediction mode, the residual signals still have different statistics. Hence, multiple pairs of transform matrices[23][27] [29] are used for a single intra prediction mode as an enhancement to MDDT. Based on KLT, each pair of matrices is off-line trained from a subdivision of mode-dependent residual signals. In addition, DCT could be included as another option besides multiple transform matrices. Moreover, this approach can be further extended for inter block transform coding [23][27].

2) Rotational Transform: The rotational transform (ROT) [20] chooses to change the DCT basis rather than to train a new KLT basis. The energy of residual signals is generally concentrated on low-frequency bands after the DCT. Due to the consideration on complexity, the ROT works only on the corresponding DCT basis of top-left 8×8 low-frequency bands of all various partitions, excluding blocks smaller than 16×16. To fit these DCT bases to a certain directional residual pattern, the coordinate system of basis is rotated by two 3D-rotation matrices for row and column transforms, where each matrix is defined by three angles for each axis in 3D Euclidean space.

F. Quantization

One element of controlling the quantization process in AVC is the quantization weighting matrix. This matrix can be either uniquely defined and sent to the decoder as coding parameters, or substituted by a default one. To match the statistics of the transform coefficient distribution, adaptive selection of the quantization weighting matrix is proposed in [21][23].

G. In-loop Filter

In AVC, a deblocking filter is applied to each decoded picture, before it goes into the decoded picture buffer (DPB),

to reduce blocking artifacts. The filter strength is adaptively adjusted according to the boundary strength. The proposals in [18][36] simplify the deblocking filter complexity. Furthermore, adaptive loop filters (ALFs) and de-banding algorithms are introduced to improve the quality of decoded pictures.

1) Adaptive Loop Filter: ALF is applied after the deblocking filter by using the Weiner filtering technique, which is similar to that in Section V-C1, to minimize the MSE between the coded and the original pictures. How to balance the bits for representing filter coefficients and the coding performance, as well as how to make proper on-off decision of filtering remain to be research problems. Therefore, various ALF schemes have been proposed.

The main idea of Quad-tree ALF (QALF) is to signal the on-off decision of filtering through a quad-tree partition process. QALF is adopted, and improved by providing multiple filters for adaptation, as suggested by many HEVC proposals [16][19][27][29][36]. In [20], the decision partition is directly derived for each CU and the partition can be merged, and as a result only the merged level is signaled.

2) De-banding: Two de-banding processes are proposed by [20]. The first one is applied after the normal deblocking filter in which offsets are sent for each group with pixels having similar edge strengths. The other one is applied after the ALF in which offsets are sent for each pixel group categorized by luma intensities. Conceptually the first de-banding process is for retaining edge strength and the other one is for matching the probability density function between the original and the coded pictures.

H. Entropy Coding

Although CABAC is proven to be efficient in AVC, it is designed for serial processing and its context adaptive feature is based on the statistics of previously coded data. A low data throughput is unavoidable and becomes a bottleneck on handling high resolution videos. Therefore, a new design for entropy encoder should consider parallelism, load-balance and complexity/performance tradeoffs.

1) Parallel Capabilities: The parallel processing capabilities of CABAC are improved in three aspects, listed from large to small: entropy-slice-level, syntax-level and bin-level parallelism.

The entropy slice-level parallelization [39][40] splits a frame into several interleaved entropy slices, which maintain and update their own context model (see Fig. 7 (a)). Slice 0 should be encoded at least one block earlier than Slice 1, so that the reference blocks for the current block (the white block) are always available.

Syntax-level parallelization [39] optimizes the loadings between multiple and independent entropy coders. It classifies all the syntax elements into several groups depending on the degree of parallelization. Since the syntax bit-rate varies for different QPs, the classification is adaptive to the QP value to keep a good load-balance.

Bin-level parallelization [16][41] is a concept of pipelining the coding process of incoming bin sequences. As shown

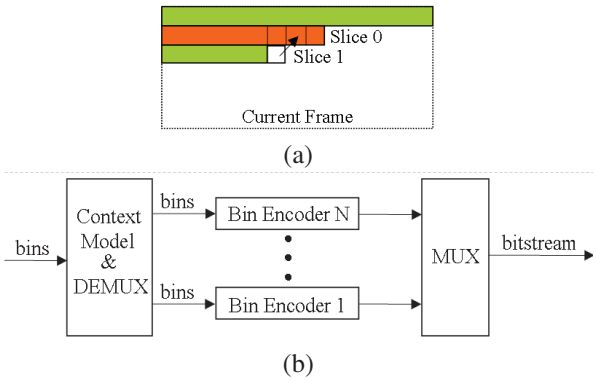

Fig. 7. (a) An example for the interleaving of two entropy slices [39]. (b) The bin-level parallelization [16][41].

in Fig. 7 (b), the incoming bins, which are binarized syntax elements, are demultiplexed and fed into one of the bin encoders based on the selected context model. The bin encoders encode bins into codewords, which will be then multiplexed into a bitstream. Moreover, the bin encoder can be implemented by any entropy encoding methods. For example, the probability interval partitioning entropy (PIPE) coder [16] uses V2V coding (see Section V-H3) as its bin encoders. And experiment shows a significant time saving while only a negligible additional overhead is paid as compared with CABAC.

2) CAVLC: In AVC baseline profile, the non-residual information is coded by the Exp-Golomb entropy coder which has no context adaptive features. Therefore, [18] proposes a CAVLC design for both residual and non-residual information with two major features. One is to improve the coding efficiency by providing more VLC tables. The other is to improve the context adaptivity by maintaining a sorting table. In CAVLC, each input is associated with a code number, which decides the corresponding VLC table. The sorting table is updated according to the probability distributions of code numbers on the fly. So, the frequent code numbers have VLC tables with shorter codewords.

3) Variable-to-Variable Length (V2V) Coding: Compared with the binary arithmetic coding, the V2V coding [16][41] has the advantage of low complexity without losing coding performance. Since a bin, 0 or 1, is assigned with a probability from the context model, a corresponding Huffman table for variable-length sequences is constructed dynamically. Then, the coding process is only a matter of table look-up.

4) Adaptive Coefficient Scanning: The quantized coefficients for each TU are zig-zag scanned in AVC for the proceeding entropy coding. To better represent of the locations of the non-zero coefficients, multiple pre-defined scanning orders are provided for selection in [17][19][20][21].

I. Increase Calculation Accuracy

The internal bit-depth increasing (IBDI) [21][26][27] increases the calculation precision during the coding process, aiming at reducing the rounding errors in intra prediction, transform and in-loop filtering. For the same purpose, [20]

identifies the minimum and the maximum values (Max, Min) of pixels in each slice, then the range [Max, Min] will be scaled to a fixed larger range during the coding process.

VI. Concluding Remarks

Several international video standards have been developed in the past two decades. They are all based on the motion-compensated transform coding framework. Several attempts have been made to invent new coding structures. During the MPEG scalable coding competition in 2004, the interframe wavelet schemes were suggested. Although the wavelet-based schemes have a more flexible scalable coding capability, its visual quality is slightly inferior. In the past 8 or so years, many researchers look for alternative coding schemes (other than the hybrid coding scheme); however, for compressing natural images, the old motion-compensated transform coding structure could still be improved and stood out in competition. A clear cost of the improved performance is the huge computational complexity.

Recently, the compressive sampling (or compressed sensing) [42] got a lot of attentions. This technique has shown some advantages in image recognition and reconstruction. However, its advantages in image compression are still under investigation. Do we hit the Shannon limit in terms of image/video coding? Is HEVC the end of video compression research? Many researchers should be interested in knowing the answers.

Acknowledgement

This work was supported in part by the NSC, Taiwan under Grants 98-2622-8-009-011 and 98-2219-E-009-015.

References

[1] H.-M. Hang and J. W. Woods, *Handbook for Visual Communications*. Academic Press, 1995.

[2] K. R. Rao and J. J. Hwang, *Techniques and for Image, Video, and Audio Coding*. Prentice-Hall, 1996.

[3] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering*. CRC Press, second ed., 2008.

[4] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Wiley, 2003.

[5] C. SGXV, "Description of Reference Model 8 (RM8)," *Working Party XV/4, Specialists Group on Coding for Visual Telephony*, June 1989.

[6] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp. 688–703, July 2003.

[7] "Report of the Formal Verification Tests on AVC," *ISO/IEC JTC1/SC29/WG11, MPEG03/N6231*, December 2003.

[8] "JM11.0KTA2.7." http://iphome.hhi.de/suehring/tml/download/KTA/.

[9] "Vision, Applications and Requirements for High-Performance Video Coding," *ISO/IEC JTC1/SC29/WG11, MPEG09/N11096*, July 2009.

[10] J. Ostermann and M. Narroschke, "Draft Requirements for Next-Generation Video Coding Project," *ITU-T Q.6/SG16, VCEG-AL96*, July 2009.

[11] "Joint Call for Proporsals on Video Compression Technology," *ISO/IEC JTC1/SC29/WG11, MPEG09/N11113*, January 2010.

[12] S. Pateux, "Tools for proposal evaluations," *ISO/IEC JTC1/SC29/WG11, JCTVC-A031*, April 2010.

[13] "Report of Subjective Test Results of Responses to the Joint Call for Proposals (CfP) on Video Coding Technology for High Efficiency Video Coding," *ISO/IEC JTC1/SC29/WG11, MPEG10/N11275*, April 2010.

[14] "Test Model under Consideration," *ISO/IEC JTC1/SC29/WG11, JCTVC-B205*, July 2010.

[15] F. Bossen, "Common test conditions and software reference configurations," *ISO/IEC JTC1/SC29/WG11, JCTVC-B300*, July 2010.

[16] M. Winken and et al., "Description of video coding technology proposal by Fraunhofer HHI," *ISO/IEC JTC1/SC29/WG11, JCTVC-A116*, April 2010.

[17] T. Davies, "BBC's Response to the Call for Proposals on Video Compression Technology," *ISO/IEC JTC1/SC29/WG11, JCTVC-A125*, April 2010.

[18] K. Ugur and et al., "Description of video coding technology proposal by Tandberg, Nokia, Ericsson," *ISO/IEC JTC1/SC29/WG11, JCTVC-A119*, April 2010.

[19] M. Karczewicz and et al., "Video coding technology proposal by Qualcomm Inc.," *ISO/IEC JTC1/SC29/WG11, JCTVC-A121*, April 2010.

[20] K. McCann and et al., "Samsung's Response to the Call for Proposals on Video Compression Technology," *ISO/IEC JTC1/SC29/WG11, JCTVC-A124*, April 2010.

[21] T. Chujoh and et al., "Description of video coding technology proposal by TOSHIBA," *ISO/IEC JTC1/SC29/WG11, JCTVC-A117*, April 2010.

[22] "JM16.2." http://iphome.hhi.de/suehring/tml/download/.

[23] H. Yang and et al., "Description of video coding technology proposal by Huawei Technologies and Hisilicon Technologies," *ISO/IEC JTC1/SC29/WG11, JCTVC-A111*, April 2010.

[24] K. Sugimoto and et al., "Description of video coding technology proposal by Mitsubishi Electric," *ISO/IEC JTC1/SC29/WG11, JCTVC-A107*, April 2010.

[25] A. Ichigaya and et al., "Description of video coding technology proposal by NHK and Mitsubishi," *ISO/IEC JTC1/SC29/WG11, JCTVC-A122*, April 2010.

[26] K. Chono and et al., "Description of video coding technology proposal by NEC Corp.," *ISO/IEC JTC1/SC29/WG11, JCTVC-A104*, April 2010.

[27] I. Amonou and et al., "Description of video coding technology proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor," *ISO/IEC JTC1/SC29/WG11, JCTVC-A114*, April 2010.

[28] T. Suzuki and A. Tabatabai, "Description of video coding technology proposal by Sony," *ISO/IEC JTC1/SC29/WG11, JCTVC-A103*, April 2010.

[29] Y.-W. Huang and et al., "A Technical Description of MediaTek's Proposal to the JCT-VC CfP," *ISO/IEC JTC1/SC29/WG11, JCTVC-A109*, April 2010.

[30] B. Jeon and et al., "Description of video coding technology proposal by LG Electronics," *ISO/IEC JTC1/SC29/WG11, JCTVC-A110*, April 2010.

[31] S. Kamp and M. Wien, "Description of video coding technology proposal by RWTH Aachen University," *ISO/IEC JTC1/SC29/WG11, JCTVC-A112*, April 2010.

[32] J. Lim and et al., "Description of video coding technology proposal by SK telecom, Sejong Univ. and Sungkyunkwan Univ.," *ISO/IEC JTC1/SC29/WG11, JCTVC-A113*, April 2010.

[33] Y.-W. Chen and et al., "Description of video coding technology proposal by NCTU," *ISO/IEC JTC1/SC29/WG11, JCTVC-A123*, April 2010.

[34] J. Ostermann and M. Narroschke, "Motion compensated prediction with 1/8-pel displacement vector resolution," *ITU-T Q.6/SG16, VCEG-AD09*, October 2006.

[35] F. Wu and et al., "Description of video coding technology proposal by Microsoft," *ISO/IEC JTC1/SC29/WG11, JCTVC-A118*, April 2010.

[36] H. Y. Kim and et al., "Description of video coding technology proposal by ETRI," *ISO/IEC JTC1/SC29/WG11, JCTVC-A127*, April 2010.

[37] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra Prediction by Averaged Template Matching Predictors," *IEEE Proc. of Consumer Communications and Networking Conference*, January 2007.

[38] Y. Ye and M. Karczewicz, "Improved H.264 Intra Coding based on Bi-directional Intra Prediction, Directional Transform, and Adaptive Coefficient Scanning," *IEEE Int'l Conference on Image Processing*, December 2008.

[39] M. Budagavi and et al., "Description of video coding technology proposal by Texas Instruments Inc.," *ISO/IEC JTC1/SC29/WG11, JCTVC-A101*, April 2010.

[40] A. Segall and et al., "A Highly Efficient and Highly Parallel System for Video Coding," *ISO/IEC JTC1/SC29/WG11, JCTVC-A105*, April 2010.

[41] D. He and et al., "Video Coding Technology Proposal by RIM," *ISO/IEC JTC1/SC29/WG11, JCTVC-A120*, April 2010.

[42] V. K. Goyal, A. K. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *IEEE Signal Processing Magazine*, vol. 25, pp. 48–56, March 2008.