Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
4th Meeting: Klagenfurt, Austria, 22-26 July, 2002

Document: JVT-D001d0
Filename: JVT-D001d0.doc

Title:	**(Draft) Report of Klagenfurt Meeting**		
Status:	(Draft) Output Document Approved by JVT		
Purpose:	Report		
Author(s) or Contact(s):	Gary Sullivan, Rapp. | Chair Ajay Luthra, Assoc. Rapp. | Co-Chair Thomas Wiegand, Assoc. Rapp | Co-Chair	Email:	garysull@microsoft.com wiegand@hhi.de aluthra@motorola.com
Source:	JVT		

---

1. Administrative Issues 2
2. Editing Progress 2
3. Ad Hoc Reports 3
4. Contributions on Draft Technical Content 3
   4.1. Motion Interpolation 3
   4.2. Motion Vector Coding 7
   4.3. CAVLC 7
   4.4. CABAC 9
   4.5. Intra Coding 11
   4.6. Lossless residual coding 13
   4.7. Error Resilience: 14
   4.8. FMO 15
   4.9. ABP 15
   4.10. SP Pictures 17
   4.11. Motion Vector Prediction, Direct Mode, & Skip Mode 17
   4.12. Loop Filter 21
   4.13. ABT 23
   4.14. Interlace Handling 23
   4.15. High Level Syntax & Timing 26
   4.16. Relation to Systems 37
   4.17. Profiles & Levels & Limits 39
   4.18. Supplemental Enhancement Information 43
5. NB Ballot Comments 47
6. Performance Analysis & Demonstration 47
   6.1. Test Model, Reference Encoding, Error Handling 47
7. Contributions on project planning 49
8. IPR-Related Contributions 49
9. Wrap-up of meeting 50
   9.1. Changes Adopted from CD to FCD 50
   9.2. Changes to Test Model Reference Encoding 50
   9.3. Further Study Areas After Meeting 50

9.4. Future planning 51
9.5. Output document review 51
10. Primary Meeting Results 51
10.1. Communication Practices, Ad Hoc Groups Established, and AHG Charters 51
10.2. Summary Information Report and Action Requests from the JVT 56

1. Administrative Issues

JVT-D-TD00r0 Info [Sullivan+] Invitation to the Meeting
JVT-Dxxx.dot Info [Sullivan+] Document Template
JVT-D000 Info [Sullivan+] List of Documents
JVT-D001 Report [Sullivan+] Report of Klagenfurt JVT Meeting (#4)
JVT-D002 Report [Sullivan+] Report of Fairfax JVT Meeting (#3)
JVT-D003 Report [Sullivan+] List of Klagenfurt Participants
JVT-D004 Report [Sullivan+] List of JVT Experts

2. Editing Progress

JVT-D015 Report [Weigand] Editor's Proposed Text Modifications
Editor's draft submitted 30 days in advance. Advance review considered sufficient for this.

JVT-D017 Report [Wiegand] Editor's proposed revision of CD relative to JVT-D015d5

Additional draft JVT-D017 containing additional changes. Some restructuring, e.g., entropy coding specification is a separate clause 10. All figures drawn in Visio format.

Comment: Reference index entropy coding should be me(v) (or something) rather than ue(v).
Comment: Avoid use of "…" in entropy coding specification.

JVT-D017d1 adopted as starting basis of further work.

JVT-D128 P(Ed.) [Sullivan] Editorial Simplif. of NAL to EBSP Syntax Interface
This contribution proposes an editorial change to the interface between the NAL unit and EBSP syntax. In the current syntax tables, there is reported to be some difficulty clearly showing the interface between nal_unit() and rbsp_extraction(). This contribution proposes a way to address this issue. We attribute the idea for the method of clarification to Mike Nilsson.

Adopted.

3. Ad Hoc Reports

JVT-D005 Report [Sullivan] AHG Report: JVT Project Management
JVT-D006 Report [Wiegand+] AHG Report: Text & S/W Editing

JVT-D007 Report [Wedi] AHG Report: Motion Interpolation
JVT-D008 Report [Horowitz] AHG Report: Complexity
"This document contains a report of the activities of the complexity minimization AHG. In addition, complexity related contributions adopted in Fairfax are summarized and reflector communications and input contributions to Klagenfurt are also discussed briefly."

JVT-D009 Report [List] AHG Report: Deblocking
JVT-D010 Report [Hannuksela] AHG Report: High-Level Syntax
JVT-D011 Report [Sullivan+] AHG Report: Timing & Timing Information
JVT-D012 Report [Marpe] AHG Report: CABAC
"This report briefly summarizes the CABAC related activities since the Fairfax meeting and provides information about the CABAC related contributions for the Klagenfurt meeting."

JVT-D013 Report [Kim] AHG Report: JM Reference Encoding
JVT-D014 Report [Wien] AHG Report: Adaptive Block Transforms
JVT-D024 Report [Karczewicz] AHG Report: Intra Coding

4. Contributions on Draft Technical Content

4.1. Motion Interpolation

4.1.1. Further Study Topic: Memory Bandwidth

JVT-D029 P2.2.1/3.1 [Hallapuro+] 4-tap Filter for Bi-predicted Macroblocks
"This contribution proposes a simplification to the subpixel interpolation of bi-predicted macroblocks. Image quality stays at the same level with the current 6-tap filter, but complexity is significantly reduced due to shorter interpolation filter."

Propose to use 4-tap rather than 6-tap filter for Bi-predictive MBs. Cuts multiplies by about 30%. Tested for common CIF/QCIF conditions and lower QP: performance neutral to PSNR (using 6-tap for the estimation process). Instruction caching issue?

Not cross-verified but software is available.

DEMO: Similar quality. No visible loss in quality (Proposal produced slightly better results for M&C).

To be investigated further as part of reduced-memory-access motion comp study.

JVT-D106 P2.2/3.1 [Benzler] Block Boundary Mirroring for sub-pel interpolation
"For motion compensation, the subpel values have to be generated by an FIR interpolation filter. The efficient MPEG-4 method for reducing the amount of reference memory access is proposed."

Memory access reduced to that of bi-linear. 5x5 region to pull in rather than 9x9 for a 4x4 region. 17x17 vs. 21x21 for

Results incomplete. QCIF results are complete. CIF not yet. Performance loss a couple of percent. Visual quality? High-res results? Think would be less impact. All picture types?

In the encoder, not using the mirroring in the decisions. Encoder search one way and then filter differently after selection of MV.

Benefits? Depends on scattering, caching issues, how the filter changes at the edge would be implemented. Would probably help substantially in hardware, but not directly proportional to the pixel count ratio (not readily quantifiable). More of a hardware implementation issue. Breaks SIMD in some designs.

Carry-over concept in 4-tap filter case.

Seems premature for adoption.

JVT-D080(r1) **P2.2/3.1 [Sato+] Adaptive MC Interpolation for Complexity Reduction**
"We previously proposed adaptive motion compensation filter for decoder complexity reduction. It was pointed out that our proposed method may cause significant increase in encoder complexity for software implementation. In this proposal we show the motion estimation and compensation method in the encoder for software implementation which will introduce ignorable loss in coding efficiency."

Current design has 6-tap filters for quarter-sample motion and 8-tap filters for eighth-sample motion. Interaction with block size issue discussed. Propose shorter filters for smaller blocks & for B pictures. Replace funny position with central position in B pictures for small block sizes when FIR3 used. Three variants of proposal described, among which recommend:

P: FIR1 for larger than 8x8, FIR2 for smaller
B: FIR2 for larger than 8x8, FIR3 for smaller

FIR1 = {1 -5, 20, 20, -5, 1} (current half-pel position filter).
FIR2 = {-1, 5, 5, -1}.
FIR3 = {1,1}.
Propose at least limiting like this for large picture sizes.

Software availability? Yes.
Verification? Not yet.
Common conditions results? Approx main profile (CABAC on, 1 or 5 ref pics, ¼ pel, +/-16 range), Interlace JM2.3.

Essentially no R-D loss reported.

Comment: Note that would still need to be able to use current filter for whole picture (if 16x16 used everywhere). Reply: The critical issue is the memory access for smaller block sizes.

DEMO

Could not see the difference between the two. However, without the side by side demo group was not willing to conclude. Too small (16) search range used on the sequences with large motion.

Visual results inconclusive. For further study in motion comp memory access AHG.

4.1.2. Further Study Item – Adaptive Motion Interpolation

JVT-D052 P2.2.1/3.1 [Wedi] Adaptive Interpolation filter with reduced complexity
"In this proposal an adaptive interpolation scheme is presented. This interpolation scheme is based on filter coefficients that are adapted once per frame to the non-stationary statistical properties of the video signal. The filter-coefficients are coded and transmitted. Compared to the results presented at the last meeting, the following modifications are introduced: The coefficient representation is reduced from 12 bit to 8 bit. The interpolation calculation is reduced from 32 bit to 16 bit accuracy. B-frames are used."

BDBR savings:

Sequence	1 reference frame		5 reference frame	
	AIF 8-tap	AIF 6-tap	AIF 8-tap	AIF 6-tap
Mobile & Calendar	7.524	5.298	6.728	4.811
Flower Garden	3.450	2.496	3.757	3.280
Tempete	1.854	1.296	2.270	1.985
Paris	1.028	1.104	1.111	0.953
Waterfall	5.678	6.290	6.228	6.647
Waterfall (no B-frames)	14.872	14.981	12.313	13.102
Foreman	2.014	2.029	2.530	2.571
Foreman (no B-frames)	6.316	5.866	5.477	5.312

Difference to previous proposals:
- 8 bit coefficients
- 16 bit implementation requiring clipping (1-~6 times per pixel)
- Results with B pictures

Cross verification via JVT-D016r1.

For QCIF no gain. For CIF, 1-6%. (13% for Waterfall sequence)

Discussion: support / error resilience problem, better understanding, not possible with shifts and adds
Conclusion: complexity does not appear to justify gains.

JVT-D016 P2.2.1/3.1 [Song+] Complexity Reduction of Adaptive Motion Interp. filter
Verifies JVT-D052 and offers fast method.

JVT-D078 P2.2/3.1 [Miyamoto+] Modified Adaptive Interpolation Filter

"This contribution proposes an adaptive interpolation filtering(MAIF) for improving MC coding efficiency. MAIF is an integrated method of AMIMB(JVT-C40) with AIF(JVT-C59). It improves coding efficiency up to 13-18% (it can compare with 1/8MC.) The additional complexity is about 5-16% of JM2 decoder and half of 1/8MC's."

- Switching of filter on macroblock basis
- P picture coding only
- No B picture results available
- No cross verification

Subjective results for Waterfall and a fading sequence shown
clear visible improvement

Gains for test set: MAIF 6-tap: 3.1%, MAIF 8-tap: 5%
Gains for set of CIF sequences: MAIF 6-tap: 7.7%, MAIF 8-tap: 10%
Gains for set of fading sequences: MAIF 6-tap: 35.7%, MAIF 8-tap: 36.3% (not considering ABP)

Discussion: Same problem with 16 bit implementation as for JVT-D052
Conclusion: complexity does not justify gains (&No B picture results available, No cross verification)

4.1.3. New Complexity Reduction Proposals

JVT-D119 P2.2.1/3.1 [Kim+] Fractional Pel Interpolation for 16-bit Operation
Simplification of motion comp interpolation method. Refers to JVT-C037 (Bossen) previous analysis. Three proposed methods to simplify the generation of "b" central (1/2,1/2) positions. Personally likes the third of the three proposed methods. Try to quantify the complexity reduction. Slight average bit rate increase for two of the three proposed methods – may just be random effects.

No action taken.

JVT-D109 P2.0/3.1 [Yamada+] Performance Evaluation on Funny Position
" This contribution proposes to remove Funny Position feature from the current CD specification based on its performance verification results using the latest JM software. The purpose is to simpify the implementation of JVT's MC process." Not present. Subjective results requested.

DEMO,

Foreman, M&C with and without B. M&C looked slightly better without funny position.
Qp=16. When looked frame by frame on some frames funny position gave smoother picture. However, without funny position showed some more details. Most people thought that slightly better results without funny position.

In demo, funny position did not appear to be better. Using the common conditions. Demo showed two sequences: Foreman and Mobile both at QP=16 (old QP definition).

Expect demos at next meeting – if it doesn't help we will rip it out. If things look the same in Geneva as they did here, it will be removed

JVT-D110 P2.2/3.1 [Sekiguchi+] Block-Adaptive Motion Comp. for complexity Reduction
"This contribution proposes a normative block-size adaptive MV accuracy decision mechanism, which reduces memory bandwidth and computational complexity during MC process while maintaining overall coding performance."

Proposal: Use of ½ - pel MC accuracy for blocks smaller 8x8 samples
Uses different (bilinear) interpolation for blocks smaller 8x8 samples in motion estimation as well as compensation
No verification
Subjective results available.

DEMO Very small difference. Some people thought CD had slightly better quality. Couple of people thought that proposal had slightly better quality.

Consider as part of reduced memory access motion comp future work.

4.2. Motion Vector Coding

4.2.1. New Improvement

JVT-D051 P2.2/3.1 [Suzuki+] Simplification of adaptive MV coding for very low bitrate
"In the Joint Committee Draft, quarter sample interpolation and one-eighth sample interpolation are supported. However, the coding of motion vectors with this high resolution may not be appropriate for the very low bitrate sequence as used a larger QP than 30 from the viewpoint of rate-distortion performance. For studying on this concern, we revised and simplified the adaptive motion vector coding of JVT-B115 for very low bitrate coding. The proposed scheme provides the improvements of 5% on average with a maximum of 10% without the serious coding loss at the medium bitrate."

Results: QP 16-28: 0.16 %, max: 2.3%, QP 30-39: 5%, max: 10%
Discussion: Gains are at very low qualities and do not justify change of the spec

The usefulness of the largest QP values in the specification was questioned.

4.3. CAVLC

4.3.1. Clean-up

JVT-D034 P2.2.1/3.1 [Au] Complexity reduction for CAVLC
"3 minor modifications to the coefficient level CAVLC are proposed to reduce complexity by 25% while enhancing compression performance: 1) Remove 19-bit escape codes from Level VLC1-4 tables 2) Improve and simplify Level VLC table selection process 3) Extend number of Level VLC tables with no additional complexity to enhance robustness to different nature of clips and higher bit rates."

Notes: Verified by Real Networks.

Recommendation: Adopt

JVT-D036 P2.2.B/3.1 [Lillevold+] Proposed updates to CAVLC
" The tables used for CAVLC have been updated in a semi-structured manner such that each entry can be stored as a 4 bit codeword + 4 bit for length (or less). This means that even the simplest implementation need only 410 bytes to implement CAVLC. The tables were also re-trained with focus on the QP range 16-28 instead of 0-28. This provided up to 1% compression improvements in this range, with little change for the low QP range"

Notes: UB Video verified the proposal. Reduces the memory size from 1.5 K Bytes to 400 Bytes on DSP architecture (verified by UB Video). Also verified by Intel (see JVT-D105).

Recommendation: Adopt

JVT-D083 P2.2.1/3.1 [Adachi+] Structured VLC based on Golomb code for CAVLC

Provides structured VLC. Models PDF with fewer parameters -> wider usage and extensibility. Does not require table storage. Either calculate the tables or pre-expand them. Creation of table require division by only 2,3 and 4.

Proposal: Replace current NumCoef/Trailing1s, Total Zeros and Run Tables. Level tables are not replaced. No significant loss (loss < 1.5%) in efficiency from current CAVLC. Sometimes gains up to 1% in Intra.

Conclusion: Not a consensus for adoption - benefits are not clear and causes (small) loss in quality. Increased complexity for 16 bit arithmetic processing. Cross verification to be done.

Recommendation: Contributors to conduct off-line discussion with people who have concerns. If no consensus can be reached at this meeting then consider setting up an ad-hoc group to discuss it further and finalize in October meeting.

Update: 16 bit issue now modified. Software not available.

Comment: There is something that works now, does not give any gain (gives avg loss in common conditions). Evaluate in AHG for October.

4.3.2. Analysis

JVT-D105 Info. [McVeigh] Cross-verification results for CAVLC
" Cross-verification results are provided for CAVLC (as described in the CD) and CAVLC with modified tables (as described in JVT-D036). These results are consistent to those provided within JVT-D036."

JVT-D036 was adopted.

4.4. CABAC

4.4.1. Clean-up & Harmonization

JVT-D019 P2.0/3.1 [Marpe+] Proposed editorial changes and cleanup of CABAC text
"This document contains a proposal for an improved description of CABAC. In addition, some technical changes are proposed that provide an improved cleanliness of the design, a significant complexity reduction and a slight overall improvement in coding efficiency."

Notes: Clean up. Editorial changes: reorganized the CABAC decision, removed inconsistency in the text description and improved clarity. Bug fix – currently no support for abp_coeff_idx.
Recommendation: Adopt

Technical Changes:

Simplification (initialization) resulting into lower memory requirement and increases coding efficiency. **Agreed and Adopted.**

CD provides the bound on C that is very low and causes loss in coding efficiency. Alternative proposal – provides no burden on decoder (other than trailing stuffing bits to throw away), does not cause loss in coding efficiency. **Recommendation: Adopted subject to cross-verification as decoder complexity reduction / clean-up.** (Removal of need for counter mechanism in decoder.)

JVT-D021 P2.0/3.1 [Heising+] CABAC and ABT
"This document contains a proposal for a simplified CABAC coding scheme by supporting only one texture entropy coding method for both modes, with and without ABT, i.e. dropping the old run and level based method
for ABT and use the new one instead. This not only leads to a unified coefficient entropy coding approach for CABAC but also to an improved coding efficiency. The new CABAC scheme proposed for ABT leads to a average reduction of the bitrate of ca. 1.3 % for INTER coding and 3.6 % for INTRA coding (QCIF,CIF test set)."

Recommendation: Adopt

4.4.2. Further Study Area: CABAC and Slices

JVT-D020 P2.0/3.1 [Schwarz+] CABAC and slices
"We propose two minor modifications to CABAC. The first modification concerns the end-of-slice signalization in connection with CABAC. In this regard we propose to replace the transmission of the number of slice macroblocks as part of the slice header by a new syntax element in the macroblock layer. This concept is considered as simplification since the encoder has no longer to buffer all encoded macroblocks of a slice until the slice header can be transmitted. If a picture is transmitted in several slices, also a small coding gain is achieved at very low bit-rates.
The second modification relates to the initialization of the CABAC context states. We propose to define three initialization tables for P- and B- slices, and another initialization table for I-slices. For each P- and B-slices, one of the three tables is selected for initializing all context states at the beginning of slice encoding. The index (0 to 3) of the chosen table is transmitted as part of the slice header. Our simulation results show that the bit-rate savings of CABAC in comparison to VLC-based entropy coding are increased by further 0-3% (2% on average) for interlaced television sequences at low bit-rates. These additional gains provide overall bit-rate savings for CABAC in comparison to UVLC of about 15%-18% for this type of sequences."

The second modification relates to the initialization of the CABAC context states. We propose to define three initialization tables for P- and B- slices, and another initialization table for I-slices. For each P- and B-slices, one of the three tables is selected for initializing all context states at the beginning of slice encoding. The index (0 to 3) of the chosen table is transmitted as part of the slice header. Our simulation results show that the bit-rate savings of CABAC in comparison to VLC-based entropy coding are increased by further 0-3% (2% on average) for interlaced television sequences at low bit-rates. These additional gains provide overall bit-rate savings for CABAC in comparison to UVLC of about 15%-18% for this type of sequences."

Interlaced test set common conditions results. New field in header, 2% improvement on average.

Notes: Proposals (1) End of slice signaling : **Adopted**
(2) Initialization of CABAC context states: **Not Adopted – future investigation.**

4.4.3. Analysis

JVT-D114 Info. [Bossen] Add. Results JVT-C038 (Arith coding complexity bound)
This document reports on the effect of JVT-C038 on coding efficiency when considering modifications to CABAC adopted at the Fairfax meeting. Results are based on the JM19_new_CABAC_020603 software which was provided by Detlev Marpe. It is noted that, as predicted in JVT-C038, the introduction of a direct mode with CBP=0 in B-frames dramatically reduces the bit overhead, especially at low bit rates. Indeed coding efficiency is not affected at all by C038 in the vast majority of cases.

Regarding: There was a previous question regarding coding efficiency penalty at low bit rates – there was no direct mode with CBP=0 in the previous software – with the new software there is no penalty for the test set.

Question: Is this a verification of complexity reduction? Not intended to have an effect on typical cases. This increases complexity in the average case. It is intended as a worst-case complexity reduction – not a typical-case complexity reduction. A profile defines a maximum bit rate, this is designed to reduce the maximum amount of average decoding symbols (and thus work) per decoded bit. Without this "governor" the codec produces up to 27 binary symbols per decoded bit in intra coding. A number of bins are not adapted in inter coding, and are thus not affected by this process. The governor reduces the 27 to 4 on average.

See also notes regarding JVT-D019.

JVT-D088 Info. [Tabatabai+] Verif. for JVT-C038 (Arith coding complexity bound)
"We report the results of cross-verifying JVT-C038, bounding arithmetic coding complexity, using the implementation provided by NTT DoCoMo. The results match those reported in JVT-D114."

Similar to JVT-D114 in content.

4.5. Intra Coding

4.5.1. Further Study Area: Complexity Reduction

JVT-D025 P2.2.1/3.1 [Karczewicz+] Analysis and Simplification of Intra Prediction
"This contribution proposes a simplification to the (de)coding of intra prediction modes. Memory requirements for the decoding process are lowered by substituting the 10x10x9 byte intra mode table with a very simple set of rules. The proposed clean-up has also a slight positive impact on coding efficiency."

Concern is quantity of table storage data for intra mode information. CD design uses 10x10x9 and 9x9 tables = 981 bytes. Proposed method: predict most probable mode with a formula based on two modes – renumbering those modes, select whether to do that with one bit, if that bit is zero, then send a 3-bit FLC to select the mode. Also tested with CABAC – similar trick and simplified contexts from 18 to 2 contexts with performance improvements.

Common conditions tested with slight performance improvement.

Also verification of JVT-D059 harmonized with this proposal.

Some similarity to Sharp proposal – main difference is signaling of selection among remaining 8 modes. Verification of the Sharp proposal. Agreement that the JVT-D025 signaling is better. These results verified by Real Networks.

New design solves an acknowledged memory consumption problem, is "cleaner", and provides better performance, and is verified.

Adopted.
We should also consider the status of ABT entropy coding. ABT entropy coding (note in JVT-D025 review) – CABAC harmonization seems resolved.

Regarding the CABAC impact – future consideration. Remark: Should consider the frequency of use of some contexts in CABAC – some contexts seldom used.

JVT-D027 P2.2.B/3.1 [Sun+] Intra Prediction Mode Ordering and Coding
"An intra-prediction-mode ordering function is proposed to enable the removal of the current 10x10x9 LUT from CD. There is no syntax change required by the proposal. The impact on coding efficiency is negligible (0.17%)."

Renumbered the modes, using minimum function to generate prediction. Supports adoption of JVT-D025 design.

JVT-D026 P2.0/3.2(JVT-D025,JVT-D059) [Zhou] Intra Prediction with Simplified Prediction Modes

Proposal to reduce number of reference samples used in intra prediction. There was a previous related proposal JVT-B033. Allows some reordering of decoding process. Friendlier to encoders for prediction decision (allowing to be done two blocks at a time). Reduces need for checking of presence of neighbor blocks.

Remark on need for clarification of order of which blocks are decoded first in current CD. This needs to be clarified. Agreed.

Remark on visual artifacts with all-Intra coding. Another remark: this could be an artifact of mode selection.

Common conditions separated into QP ranges +/-3%, avg. 0.3% loss. Test results are not a test of this idea alone, but changing both the mode selection (somewhat similar to the Nokia proposal) and also the reduction in number of samples. Outside of common conditions about 1% loss average with static background. Visual demos.

Vertical aspects especially interesting.

Visual demos available.

Further study on memory access issue for October. Create AHG on this.

JVT-D059 P2.2.B/3.1 [Conklin] Low Complexity Diagonal Mode for Intra Prediction
"A new definition of mode 7 (i.e. "horizontal up") is proposed that lowers its computational complexity, reduces the worst-case intra-prediction complexity by 1/3 and improves coding performance up to 1.6%."

CD mode 7 is more complex than the others. This proposal makes the design of mode 7 more consistent with the others. Modes 6 & 7 become transposes.
Can combine with JVT-D026? To be investigated.

Adopted.

4.5.2. Further Study Area: Chroma Prediction

JVT-D062 Info. [Suzuki+] Results of Chroma Spatial Prediction for Intra Coding
"This contribution reports the results of improved chroma spatial prediction for Intra coding. It was proposed in JVT-C118 in Fairfax to improve the coding efficiency of chroma. The test is performed using HDTV (1920x1080 30 Hz interlaced) materials. RD characteristics and the subjective quality are compared. The results will be demonstrated in Klagenfurt by D1."

Proposal JVT-C118 (Viscito) verification using HDTV interlaced material with field-structured coding. Was proposed to use DC, Horizontal, Vertical, and Plane prediction (similar to the luma 16x16 modes) – rather than just the one mode (DC).

Tested all intra and IBBP.

CABAC on, search +/- 32, R-D Opt., QP: 16, 20, 24, 28 (old QP origin).

Compare with JM2.0

Got better luma & chroma R-D performance.

Average performance:
BDPSNR: 0.1 dB better luma, 0.5 dB better based on chroma
BD-Rate: 2% savings luma, 14% savings based on chroma.

Author recommends adopting JVT-C118.

DEMO

Favored the proposal: less chroma blockiness was clearly visible. JM2.1 (no ABT) was used.

JVT-C118 method adopted.

4.5.3. Analysis

JVT-D061 Info. [Suzuki+] Study of Intra Coding in JVT
" This contribution reports the performance evaluation of Intra coding in JVT codec. The performance of Intra coding in JVT is compared with Motion JPEG2000. The test is performed using HDTV (1920x1080 30 Hz interlaced) materials. RD characteristics and the subjective quality are compared. The results will be demonstrated in Klagenfurt by D1 tape."

DEMO

With J2k VM8.6 not optimized and R-D optimized JVT JM2.1, J2k looked better had more detail in one sequence. However, new tools (like ABT) may improve the JM2.1 quality in the chroma with details. In second sequence, JVT looked better. In third, they were equal.

JVT-D039 Info. [Halbach] Performance comparison: H.26L intra coding vs. JPEG2000

4.6. Lossless residual coding

4.6.1. Further Study Area: Lossless residual coding

JVT-D028 P2.2.B/3.1 [Sun+] New Results of Lossless Coding
Bypass the transform and quantization. Current QP=2 is approximately the break-even point on compression ratio as a comparison to lossless coding on average. Did not evaluate mixing lossless with lossy in this test – this is slice-level decision not macroblock-level decision. This proposal is not motivated by limiting data expansion – it is a proposal for a truly lossless coding mode.

Comparison of JPEG-2k with VLC and CABAC: Significant gain relative to JPEG-2000 lossless.

Architectural similarity to current JVT design is part of the argument.

Version 2? Sounds very interesting as potential version 2.

JVT-D054 P2.0/3.2(JVT-D028) [Wien] Verification of Sharp Lossless Coding Proposal

" In document JVT-D028 a lossless coding mode for JM coder is presented. The work is based on the original proposal JVT-C023 from the Fairfax meeting. This document provides a cross-verification of the results presented in JVT-D028. Shijun Sun from Sharp Labs of America provided an implementation of the lossless proposal into JM20. The software was compiled and tested successfully on a Linux PC using GCC 2.95.3."

Shows essentially same results as JVT-D028.

4.7. Error Resilience:

4.7.1. New Proposal: Flexible Data Partitioning

JVT-D136 P2.2/3.1 [Ye+] Flexible Data Partitioning for Video Streaming
"We propose modifications needed to enable a two-partition mode that allows flexible splitting of AC coefficients for the streaming profile. We first describe an unequal error protection scheme that can effectively prioritize base and enhancement layer packets and how the flexible two-partition mode can be used in such video streaming systems. We then specify the small changes needed to enable this mode. Finally we present simulation results"

Proposes adding an additional data partitioning mode:

Unequal error protection has been seen as a difficult issue, but can be achieved in some environments. Wireless LANs. 802.11e provides some such features – but need to go beyond that. DIFFSERV work in IETF. Can also do UEP on application layer.

Proposing establish a "priority breakpoint" in slice header – put some coefficients of partitions A & C in the "base layer" partition and higher frequency coefficients in "enhancement partition". New partitioning is a two-partition scheme with only high-frequency coefficient. Roughly a 50/50 split between base and enhancement partition rates at QP=16 for all intra DC and intra 4x4 and inter DC in the base layer.

Proposed as offering more choice of error resilience prioritization.

Comment: Does not seem to fit with the current design of entropy coding.

Comment: Version 2?

JVT-D135 Info [Tourapis+] Verification results on flexible data partitioning
Verification with similar results to JVT-D136.

4.8. FMO

4.8.1. Future Work Item: FMO with Rectangle Support

1st part of JVT-D095 P2.2.1/3.1 [Hannuksela+] Enhancements to FMO

First idea in proposal:

Example Use Case 1: Rectangular slice groups.
Example Use Case 2: Region of interest exclusion/inclusion area (e.g., rectangular)
Example Use Case 3: Picture-in-picture, or multiple camera view (e.g., continuous-presence multipoint)

Coding these example patterns in current FMO design would take lots of bits at parameter set level.

Comment: Need to be able to change FMO maps. Generally agreed that this is needed.

Syntax consists of defining slice groups with RECT data structures, with last slice group being implied as the remaining parts of the picture.

Comment: Allow overlapping? No. Could do this with the order determining what slice has priority.

Comment: What is the rationale for having a number of different specific modes for specifying FMO slice groups? Idea is to have optimized way to express the most commonly-used styles of operation and a remaining general way to describe any other pattern.

First idea adopted.

4.9. ABP

4.9.1. Clean-Up: Eliminating/Reducing Block Weight Syntax for ABP

JVT-D122 P2.2 [Boyce] Adaptive ref. picture weighting using ref. pic. index
"JVT-C066 proposed two methods of adaptive weighting of bi-predictive pictures, implicit and explicit. In the explicit method, an adaptive weighting factor index is transmitted once per motion sub block or 8x8 region. In this contribution, a clean-up to JVT-C066 is proposed, where the reference picture index is used to select the weighting factor, rather than explicitly transmitting a weighting factor index."

There is currently an "implicit mode" based on temporal location of reference pictures.
In explicit mode, transmit a weighting index for every bi-predictive motion block, except perhaps direct mode. Some question of clarity of direct mode issue in draft – interpretation in contribution is that the weights also sent in direct mode. Propose to send a set of weights for each reference picture, and then use the same weight factor each time a particular reference picture is used.

Question: Calculation of overhead? Guess of 2 bits per index. Maybe that's a high estimate due to confusion over direct mode issue.

Huge performance improvements reported in fading, but there is some issue of how these results were computed.

Weighting factor for each forward or backward picture or for each picture pair?

Question: Does this have a divide operator? Ans. Can use a constant denominator with a shift.

Difficult to compare results.

Compromise solution drafted between JVT-D122 and JVT-C066/JVT-D089 method for consideration. JVT-D122 is viewed as finalization clean up of a new tool (ABP) with huge gain under typical use relative to JVT-C066/JVT-D089.

Why not just use JVT-D122 only? Is it clear that the flexibility is useful? Why have the extra switching and syntax for having the feature at the macroblock level?

No evidence that macroblock-level ABP is useful. Is there a need for macroblock-level ABP? Macroblock-level syntax does not appear needed – can consider as future question.

Precision: 8-bit versus 4-bit weight accuracy. 8 bits appears appropriate at this time.

Apply to chroma? Yes.

Apply to P frames? Yes. Non-baseline due to extra multiply complexity.

JVT-D122 method adopted – no macroblock-level syntax to be included at this time.

4.9.2. Analysis

JVT-D089 Info. [Kikuchi+] New result on the Adaptive Bi-Pred. Interp. (ABPI)
" This contribution reports a new simulation result of the adaptive bi-predictive interpolation using a new software based on the latest JM (JM-2.1). Apart from fade sequences, a one minute sequence containing various scenes are used with wide range of bitrate (up to 1Mbps) and image sizes (SIF to SDTV) for the evaluation in a realistic condition. The results show a superior performance in both objectively (2 to 8 dB improvement) and subjectively. D1-tape demonstration will be provided."

DEMO

During the fading porition ABP was clearly better, especially when M=1. Saw the improvement when M > 1 but less than when M=1.

4.10. SP Pictures

4.10.1. Bug Fix & Editorial

JVT-D137 Prop.(Editorial) [Nilsson] Improved Text for SP Pictures
"This contribution proposes improved text for SP pictures for the FCD, and can be considered to be addressing bug fixes for both the CD and the reference software. No new technical ideas are proposed."

Adopted.

4.11. Motion Vector Prediction, Direct Mode, & Skip Mode

4.11.1. Clean-up: MV Prediction & Direct Mode Definition

JVT-D033 P2.2/3.1 [Abe+] Clarification and improvement of direct mode
About the case when the co-located MB in the 1st backward reference pictures is bi-predictive or direct – what is the direct prediction? Not specified in the current draft.

Based on VCEG-O26 and JVT-B057.

I0, B1, B2, P3. If P3 refers to I0, prediction of B2 will not use B1 even though it has been decoded and is available. Propose using B1 rather than I0.

Propose forward reference picture for direct mode is always the picture with index 0 in the forward reference order, regardless of what picture was used for coding the co-located macroblock in the picture with index 0 in the backward reference order.

When the co-located MB has no forward reference or in intra coded, current MB is decoded assuming forward prediction with zero motion vector and reference picture index 0 in the forward reference order.

Simulation using JM2.0.

Two cases:
1) using TR
2) using explicit scaling factors as in current draft.

1st of each pair of B pictures is searched in coding of the 2nd of each B picture.
Results show improvement from the change of the direct mode in all cases. Results reported only for the B frames – should be provided for entire sequence.

The method reverts to the usual case if the 1st of each pair of B pictures is not stored.

JVT-D040 P2.2.1/3.1 [Tourapis+] MV Pred.: Time-Indep., Divide-Free, Perf. Impr. & Bug Fix
"In this contribution new methods for calculating the Direct Mode motion vector parameters are presented, by using the 16x16 Motion Vector Predictor. Our methods can achieve timestamp independent motion vector prediction, but can also considerably reduce memory and complexity for both the encoder and decoder by eliminating all divisions required by the current calculation of Direct Mode. These methods combined with a "conditional" consideration of the Non Residual Direct Mode can considerably outperform the existing JVT design. An adaptive at the slice level method that allows the combination of different Direct Mode schemes is also presented. Finally, some editorial changes are proposed to the current description of Motion Vector Prediction."

Time-related aspects:

Scenarios in current design: What to do in case of co-located MB being intra? What do you do with a scene change – do you average the scenes together in all direct-mode MBs. Proponent's view is that current temporal prediction in direct mode is broken. Using prediction of MV values based on spatial-neighbor MVs rather than based on temporal relationships is advocated.

If we want to eliminate time from the decoding process, should change the semantics of the direct mode to, instead of using temporal scaling of MV values, use prediction of MV values based on values of MVs in neighbor MBs.

Also handles interlace, because eliminates dependencies on values of MVs in other pictures.

Does not need a syntax change – just a change of method of predicting MV values.

Actually a complexity decrease because scaling of MV prediction uses division, and division operations are something we have intended to avoid.

Several variants of scheme in proposal. Focus in this discussion on the prediction method basing everything on spatial neighbors and indication of whether co-located MB is zero or not – not using time at all. Average gain over common conditions test set (although RD optimization for the prior method could perhaps be improved and if this is done the average improvement might not be as large).

Also has a scheme for using direct scaling parameters as in current CD text "TMVP". This combination scheme has better performance on average.

Also has a scheme with a flag at the slice level to select which scheme to use. This scheme has even better performance on average.

All of these eliminate time.

Software? Available.
Verification? Yes.

Clarity suggestions for P pictures: Adopted.

See remarks elsewhere regarding state of direct mode and future study.

JVT-D058 Info. [Winger] Cross-Verification of JVT-D040 Results
" Cross-verification results are provided for JVT-D040 "Timestamp Independent Motion Vector Prediction for P and B frames with Division Elimination".

Source code provided to the reflector was compiled and the "MVP" method was
run under common conditions. These results are consistent to those provided within JVT-D036."

See notes relating to JVT-D040.

JVT-D059 P2,2/3.1 [Suzuki] Handling of reference pictures and MVs for direct mode

"After the publication of JCD, some questions were raised about direct mode. The problems are classified into two issues; the position of reference pictures and the existance of forward MV in collocated block. This contribution provides the revised syntax of picture layer for clarifying the position of reference pictures and the adaptive direct mode that does not require the TR of reference pictures when they are not located in the normal position."

Change proposed to picture header, slice header, and definition of direct mode.

Prediction of MV may not be good if unconventional temporal relationship relative to reference pictures.

Solution is to send an indication of whether pictures are in "normal" temporal relationship or not.

In "normal" case: Operate essentially as in current draft. However, need weighting factors for every reference picture if co-located MB in backward reference 0 uses a reference picture other than the first forward reference. Remark: Concern expressed over amount of overhead that may be involved in this slice header information.

If co-located block does not have a forward MV, what to do: Simple solution is to set MV to zero. But this may not be a good idea.

In "non-normal" case: Direct mode operates at 4x4 level (issue of whether the 4x4 case would be allowed in B pictures based on profile & level discussions). Some similarities over how remainder of operation functions with JVT-D040 contribution (select reference picture and MV value based on values in spatial neighbors).

Implemented in software 2.1 with PSNR computation bug fix.

Some experiments with IBBIBBI, etc. case.

Common conditions sequence and some others. Average IBBI… Significantly better than current method. IBBP result a little bit of gain relative to current draft method (not relative to pre-Fairfax draft).

See remarks elsewhere regarding state of direct mode and future study.

JVT-D056 P2.0 [Jeon] Direct mode in B pictures

Issues discussed regarding temporal direction of prediction process and decoding order. Assumes display time information is available for use in the decoding process, and advocates ways to use the time information for direct mode prediction.

Performance: Not yet available.

See remarks elsewhere regarding state of direct mode and future study.

JVT-D057 P2.0 [Jeon] Clarification of MV pred. and pred. signal in B pictures

Prediction of motion vector values. Indicates clarification needed. Advocates using temporal display order information to determine motion vector prediction.

See remarks elsewhere regarding state of direct mode and future study.

JVT-D074 P2.2/3.1 [Kimata] Alternative Skip Mode in B-picture

Skip is direct in current spec. Proposal is not to change direct mode with residual – only the direct skip mode. Some similar concepts to Tourapis and Suzuki.

Test relative to JM1.9 comparing to pre-Fairfax method. IBBP, with motion copy.

5 reference pictures UVLC: Avg 3% (peak 5%), 1 ref 2% (peak 4%).
5 reference pictures CABAC: Avg 3%, 1 ref 2%.

Illustration that some gain is shown for the elements of the proposal.

Comment on interaction with R-D optimization of direct skip usefulness.

Note this proposal uses timestamps in direct mode MV computation.

How do we compare methods to each other?

Comment: Not needing to store the motion vectors is a useful feature.

At least 5 issues: Slice layer syntax, performance, division, temporal reference, memory requirements for storing motion vectors. Decision at last meeting was to not use temporal reference – we should follow through on that decision unless significant evidence that we should use it.

Question: Best method that uses TR versus Best method that does not use TR? What performance penalty?

Interim discussion on direct mode during meeting: Interim proposal summarized: basically revert to wd2 method. Add bit to slice level to select whether to use spatial-based or temporal-based direct MV calculation method. WD2 method is used if temporal. Alternative method to substitute for temporal method (Haskell & Nokia).

See remarks elsewhere regarding state of direct mode and future study.

4.12. Loop Filter

What is effect on interlaced sequences?

4.12.1. Complexity Reduction

JVT-D037 P2.2.1/3.1 [Joch] Loop Filter Simplification and Improvement

" The document presents 4 minor modifications that are intended to clean-up and simplify the loop filter design. The changes generally reduce complexity, while maintaining or improving subjective quality."

DEMO
Could see some difference in the sequences. Some people think the proposal seems to smooth out the details. For higher bit rate (2Mb) could not see the difference. Mixed opinion. Overall, demo was not conclusive.

Some of this discussed in AHG

1. Adding constraint on gap size for strong (5-tap) filtering – some problem reported as designed in CD. Adding an additional constraint to 3-tap versus 5-tap filter decision. Quality improvement. Reaction to an expression of a Fairfax problem. Some support. Verification: some verification expressed verbally. Remove a Fairfax mistake – adopt.
2. Modified tap values for P1/Q1 in strong filter. Quality improvement. Visual demo on still frame – some improvement shown. Some support. Adopt.
3. Chroma default filtering change. Complexity reduction. Dealing with previously-reported issue – flaw in filter design. Some support. Some overlap with JVT-D049. Compare to Thomson proposal in AHG.
4. Boundary strength simplification. Quality improvement (sometimes). Significant simplification if not reducing quality. See demo remarks. For future study for October. Seemed to be some quality degradation.

JVT-D038 P2.2.1/3.1 [Joch] Loop Filter Tables and Variable Indexing
"The document proposes a set of subjectively tuned the Alpha and Beta tables for the loop filter, which also fix an inconsistency that existed with the tables in the CD. Second, we propose a new method of providing encoder designers with flexibility in the properties of the loop filter by allowing a variable offset to the index used to access the tables to be transmitted in the bitstream. The method, called Variable-Shift Table Indexing (VSTI) will make the JVT standard more robust to a wide variety of content types, resolutions, and encoder characteristics, with minimal additional complexity."

1. Modification of alpha and beta tables. Doubling rate of alpha tables. Allow filter at low QP. Subjective improvement. Some support – characterized as a bug. Significant discussion & evaluation prior to this meeting. Adopted.
2. Variable-shift table indexing (VSTI). Offset value of QP. Range of values -12, -10, -8, …, 8, 10, and -36. Remark of default tables being too strong for detailed and high-res content, too weak for other sequences. Discussed in AHG. Some support – like the notion. Adopted.

Subjective improvement.

JVT-D049 P2.2/3.1 [Gomila] Adjustments for chroma deblocking

1. Simplification for chroma. Test for small gap across an edge and turn off the filtering below a certain gap value across an edge. Complexity reduction.
2. Adding an extra comparison to reduce number of "void" computations. Adds a couple of operations. Some complexity added in SIMD (but doesn't increase length of worst-case path), complexity reduced in sequential implementations.
3. Change of default filter for chroma.

Average complexity reductions reported for part of the filter: the "line-based" operations.

Early exit from line-based – does not help on media processors (any heavily pipelined processors). Average complexity reduction – what about worst case? Stated to not increase worst case.

Some loss in chroma PSNR approx. 0.1 dB average. Visual results? Common conditions? yes (progressive).

Compare to UB Video proposal in AHG. The chroma filter issue will be the only task of this AHG.

4.12.2. Clean-up

JVT-D113 P2.2 [Lim+] Filtering Strength for B Pictures
"The method to determine Boundary Strength (Bs) for blocks in B pictures is not described clearly in both the CD and reference software implementation. The purpose of this proposal is to propose a better description of the boundary strength determination method and to present results showing the improvement in subjective quality as compared to the reference software implementation."

- There appears to be a forgotten case in B picture filtering. Check for all MVs and all reference frames. Visual quality shown. Problem previously known and taken care of in software. Fix in text as clarification OK.
- 2nd issue, clarify semantics. OK (get text).

4.12.3. Further Study Area: Dither

JVT-D132 P(removal) [Syed+] Removal of Dithering from Baseline and Main Profile
"This documentation proposes that dithering in the loop filter be removed from the baseline and main profile."

Adopted. Dither to be removed entirely from the draft.

4.13. ABT

4.13.1. Analysis

JVT-D047 Info. [Baylon+] Verification results for ABT coding
"This contribution presents some verification results for JM2.1-based ABT coding of interlaced sequences. The results show significant bitrate savings with ABT coding over non-ABT coding."

Verifies previously-adopted ABT benefit.

4.13.2. Harmonization

JVT-D053/M8549 P2.0/3.2(RBosch) [Wien] Changes for ABT
"In this contribution, some clean-up for technical details and some editorial changes are proposed for the description of Adaptive Block-size Transforms (ABT) in the Joint Committee Draft. The changes relative into JVT-D017d0.doc are given in the acompanying document JVT-D053_changes_to_D017d0.doc with change marks on. The technical changes regard the integration of the useABT flag in the slice header and the harmonization of the scaling with the modified quantization range of the JM."

Proposes editorial clean-up and harmonization, adding semantics. Put bit for using ABT in GOP-level parameter set. Adopted.

4.14. Interlace Handling

4.14.1. Clean-up

JVT-D018 P2.0/3.0 [Wiegand] Multi-picture handling

Picture number assignment.
A frame is two fields. Each field has a PN. 1st field is the field with the lower PN.

Reference picture reordering
Reordering is proposed to be frame based in the case of frame coding. Comment: Problem with regard to difference of picture numbers. Apparent misconception in proposal about how abs_diff_pic_numbers works.

Buffering Management
MMCO always field-based in spirit.

New version of JVT-D018 (ppt) supplied after discussion of various aspects. New (r1) version adopted.

JVT-D086 P2.2/3.1 [Schlockermann+] MMCO for Interlaced Sequences
"Current CD does not describe MMCO commands for interlaced sequences (field processing and adaptive frame/field processing). This proposal provides MMCO commands for interlace processing."

Proposal: Buffer is a frame buffer, numbers are frame numbers. A dangling field is stored in a frame store, wasting the memory for the other field.

Introduce concept of progressive sequences? Maybe not really intended.

Sliding window is proposed to be based on frames.

Resolved by JVT-D018r1

JVT-D042 P2.2.B [Panusopone+] Colocated Blocks & Ref. Frames/MVs for Direct Mode in AFF
"This documentation provides the definition of the collocated blocks, and the associated reference frames and MVs for direct mode MB in B for AFF."

A definition of the meaning of a co-located area in the case of adaptive frame-field. Such a definition is clearly required. Part of direct mode discussion.

Adopt.

JVT-D043 P2.2.B [Panusopone+] Reference Frame Numbering and Copy Mode for AFF
"This documentation proposes two modifications for field coding of AFF. One is on reference field numbering and the second is on copy mode for P."

In CD, skip copy uses the same parity field by default. No benefit shown for changing this. Not adopted.

JVT-D044 P2.2.B [Panusopone+] Temporal Reference for AFF
"This documentation specifies temporal reference (TR) in the picture header of JVT for adaptive frame/field (AFF) coding."

We believe JVT-D017 defines temporal_reference in a manner consistent with what is requested. No change needed.

JVT-D045 P2.2.B [Panusopone+] Reference Pic. for B Pic. using field picture structure
"This documentation proposes an extension to a data-dependence order for B picture when using field picture structure. This new order allows second field of a B picture to point to the first field of the same picture."

We believe JVT-D017 defines B picture data dependence in a manner capable of the requested action. No change needed.

JVT-D046 P2.2.B [Panusopone+] Direct Prediction for P Picture in Field Coding mode
"This documentation proposes direct prediction mode for predictive (P) picture in field coding mode. This new mode is an extension of the direct prediction mode, which has been considered in JVT standard at the Fairfax meeting."

A proposed new improvement without test results. No action taken on this.

JVT-D090 P2.2.B [Panusopone+] Multi-frame interpolative prediction in AFF
"This documentation proposes a coding rule for multi-frame interpolative prediction mode for frame/field picture in AFF coding mode."

Extends ABP to field coding. Part of ABP work. Coordinated with that effort and revisited. Can be harmonized.

Harmonization to be reflected in JVT-D155.

4.14.2. Further Study: MB-Adaptive Frame/Field

JVT-D108 P2.2.B [Wang+] MB adaptive field/frame coding for interlace sequences
"This document presents the results of MB-level adaptive frame/field coding for interlaced video materials, and the performance comparisons with picture level adaptive frame/field coding. The simulation results show that MB level adaptive coding can provide additional gain over picture level adaptive coding for sequences that favor frame coding."

Got performance gain about 9% on average over a set of 6 test sequences. These are not the common conditions test sequences, but appear representative of pictures that contain a mixture of regions with differing motion characteristics.

Comparing over using all frame coding or all field coding would show very large gain.

QP values are low.

Text description?

"macroblock pairs" concept.

Potential advantage noted in regard to ease of designing encoder allowing AFF decisions to be made in the encoder without batching up results for entire picture first.
Will upload software as supplement to document.

Adopted as cleanup of adopted interlace adaptivity, as somewhat of an exception to the general policies of adoption and subject to cross-verification.

Frame deblocking is used for frame pictures – field deblocking is used for field pictures. What do we do for MB-level frame/field? Answer: Frame-based deblocking.

JVT-D081 Info. [Sato+] Exp. Results MB-level Field/Frame Adaptive Coding

Partial verification of JVT-D108.

Further Study Item: Chroma Phase Shift
JVT-D079 P2.2 [Sato+] Chroma Phase Shift for Interlace Video
"We previously proposed adaptive chroma phase shift for interlaced video with a 6-tap FIR filter. In this contribution we show that chroma-phase shift with a 2-tap FIR has a similar effect as our previous proposal. Moreover, the newly proposed method requires less memory bandwidth and comp. complexity than the previous one."

Up to 7.4% gain (was 10% in Fairfax), average 3%, down to zero in some cases on common conditions for interlace.

Software available.

Comment: How about 3-tap filter (including sample of other field)?

Some issue about assumptions the generation of the chroma samples – whether this is a better estimator depends on assumptions of how the input samples to the encoder were generated.

Proposal appears significantly below the Fairfax threshold.

There are temporal offset issues for luma as well.

Further study of these issues planned.

4.14.3. Improvement

JVT-D073 P2.2.1/3.1 [Jeon] Alternate Scan for Interlaced Coding
"The document proposes to allow other scan direction such as the alternate scan in addition to the current zig-zag scan order when the 4X4 residual transform is used. Note that more than one scan directions are already possible in current H.264 draft when the ABT is used. The experimental result with (IPPP) encoding interlaced video as the frame pictures and using the alternate scan with the 4X4 transform demonstrates that additional bit rate reduction (BDBR) of up to 8.64% and 6.15% on average is possible by only letting the alternate scan direction usable."

Comparison to AFF?

Using alternate scan all the time with interlaced pictures? Best gains appear to arise from using inappropriate picture structure for the coded sequence.

Late stage of project, lack of verified results, lack of comparison to best available encoding method, addition of new syntax element. Potential coupling of coding tool to interlace/progressive indication.

Potential harmonization with ABT design: Use alternate scan in fields always? Get results with cross-verification in October.

To investigate in ad hoc group activity.

4.15. High Level Syntax & Timing

4.15.1. New Proposal: Robust MMCO Repetition

JVT-D031 P2.2/3.1 [Kadono+] Error robustness memory management control operation
(related to JB remark)

Advocates indicating that MMCO short-term unused, long-term unused, long-term index assignment, or reset command (with indication of when reset occurred) is a repeated command. By knowing that the command is a repetition, the decoder can attempt to include the effect of the repeated MMCO command prior to decoding the current picture (rather than having MMCO affect only the buffer state after the decoding of the current picture).

Does the draft currently specify that every slice must carry the same MMCO commands? Should it?

Depends on the picture header issue.

Not affecting normative decoder behavior.

Move the repetitions to SEI.

Covered in high-level syntax output.

4.15.2. New Proposal: Slice Group Priority

2nd part of JVT-D095 P2.2.1/3.1 [Hannuksela+] Enhancements to FMO
Second idea in proposal:
Signal a relative priority for the slice groups. Could send in parameter set or SEI. Proposal puts it in parameter set.

Data is not intended for decoder consumption.

Comment: If priority is used in a gateway, the gateway would need to parse the slice headers as well to determine which slice group each slice belongs to.

Comment: Could make a priority bit in the first byte of the NAL unit.

Considered in high-level syntax output.

4.15.3. Other

JVT-D101 P2.0/3.1 [Hannuksela] Sync Pictures
Coded pictures representing the same picture contents (redundantly) are referred to in the document as sync pictures. The paper proposes slice header syntax and semantics to signal the presence of sync pictures (multiple representations of the same picture) in the data stream.

Note: This feature was present as SEI in H.263.

Desire to send different pictures with the same picture number.

If a transmission order is defined, then can say that extra pictures should be discarded.

Propose a parameter set flag indicating whether stream may contain sync pictures. In slice header, if the parameter set flag is set, add a syntax element indicating whether the slice is the primary representation (index 0), the secondary representation (index 1), etc.

Why not support "sync slices" rather than just sync pictures? Could be done.

"Subsequence" concept included – this part not relevant in the case of lack of normative subsequence support.

Can use for video redundancy coding, error resilience repeating part of pictures, etc.

Comment: Overlap in concept with "spare reference picture" concept? Could be some room for alignment of these concepts.

Mismatch issue: Can avoid with use of SP/SI pictures.

Some interplay with idea of constraints on NAL unit order. Also with timing & display issues – what's the difference between a spare reference picture and a non-displayed non-stored ordinary picture?

Interest expressed in the concept. Adopted in Baseline but not in Main.

JVT-D032 P2.2/3.1 [Kadono+] Frame Number (PN) Continuity at SP-picture
PN may be discontinuous when a stream is switched at SP-picture. It makes difficault for error detection mechanism by checking PN continuity. We propose a frame_num_s element into picture header syntax of SP-pictures.

Comment: Should PN = 0 for all SP/SI pictures? Does the draft specify this? We believe the answer is no – the draft assumes normal PN behavior in an SP/SI picture. Should it be forced to do this? - perhaps we should rely on MMCO as described below instead and leave the choice to the encoder.

Comment: When the buffer is reset with an MMCO command, can PN be reset to zero? Should that be required? Perhaps. During the decoding process or after the decoding process? After.

Either of these alternative methods would seem to enable this functionality without adding new syntax (without needing two PN's in each slice header).

Recommend PN=0 to be assigned after decoding a picture (in which the current picture PN syntax element is assigned in the ordinary) with MMCO=Reset as the way to address this issue.

Covered in high-level syntax output.

JVT-D123 P2.2 [Boyce+] Non-Stored Picture Count
"The current frame_num syntax specifies that any number of consecutive disposable pictures have the same frame_num value as the stored picture that follows them in transmission order. This does

not allow that each coded picture be uniquely identified within the VCL syntax. We propose two possible solutions to the problem."

Problem: What if two disposable pictures are sent in a row? When TR's not attached to pictures, what if receive part of the first one and part of the second one.

Possible solution: At the systems layer, provide an association of slices to distinct pictures.

Alternative solution: Add a non-stored picture count (to picture or slice header syntax) incremented for each non-stored picture.

Second topic: Propose to send PN and disposable picture count within TR SEI.

Question: How do you associate TR (or any SEI) with the data it belongs to? Remark: Transmission order is intended to define this. Is that stated in the spec? Remark: Maybe not.

Comments regarding robustness issues: Must decide how much to think about error recovery versus error concealment, etc. Need to think about the probability of various error scenarios.

Question how serious the problem is that this is trying to address.

Believe there will be some indication of a problem from the systems layer – e.g., such as packet sequence number.

Some interaction with TR-in-NAL issue.

Future AHG work planned on time and decoding process, prior contribution from Hannuksela related.

4.15.4. Further Study Item: Timing Handling

JVT-D055 Info. [Jeon] Usage of relative timing information in VCL
Relative timing information in video layer would indicate the display order of each picture and also enable computation of temporal distance between two pictures. This document advocates including relative timing information in the video layer.

1. 9.2.1.2 default index order for B pictures uses display order (9.2.1.2.1 & 9.2.1.2.2). Remark: This is a bug. There was a simple way to fix this described in ad hoc email.
2. 11.4.2 computation of the prediction of backward motion vector is based on the forward motion vector and time. Remark: This is a bug.
3. 11.5.1 implicit ABP coefficient uses display order. Remark: This is a bug.

Prediction of forward and backward MV values use time? Comment: No, time is not used for that. "Forward" and "backward" are not time-based terms.

Proposal is to provide timing information to allow these clauses to function without alteration. Remark: These are bugs. There are other ways to fix them and we should consider alternatives to

time-dependence in determining how to fix them. The stated goal coming out of the Fairfax meeting was to make the decoding of sample values not depend on time – and any remnants of time-dependence are simply bugs.

JVT-D124 P2.2.1/3.1 [Haskell+] Variable-Accuracy Inter-Picture Timing for Video
" The present contribution proposes a simple yet powerful method for transmission of inter-picture display time values. The proposed method, which addresses the case of variable inter-picture display times in video sequences, makes possible obtaining very high coding efficiency and selecting the accuracy such that it meets the requirements of the decoder.

Such a capability is potentially needed in [1] Sec. 11.4.1 & 11.4.2 to compute differential motion vectors, Sec. 11.4.3 to compute Direct Mode motion vectors and in Sec. 11.5.1 to compute Implicit B Prediction Block Weighting."

Future AHG work planned on time and decoding process, prior contribution from Hannuksela related.

Three types of delta value patterns:

- Regular
- Slightly regular
- Irregular

Does not need to be sent in baseline (only needed in B picture decoding process)
Method shown to limit division to once per picture and reference picture

Adopted without the Slightly regular. [Get clarification of what this means from Thomas]

4.15.5. Further Study: NAL, Picture Layer, Dynamic Parameter Sets

JVT-D065 P2.0/3.1 [Wenger+] Parameter Set Issues
"Proposed are a) a tagged format for some of the Parameter Set entries, b) default values for frame_cropping (to 0) and the video input parameters (to unspecified), c) a semantic definition which Parameter Set values have to stay constant in a picture/GOP/sequence, and d) a maximum number of parameter sets that have to be available in each decoder (32)."

Proposed:

- One flag (per line) that indicates the presence of:
  - frame_size (two VLC-coded values)
  - frame_cropping (four VLC-coded values)
  - aspect Ratio (data structure)
  - timing_info

Recommended for adoption. Note NOT retaining of most recently received parameter set values. Covered in high-level syntax output.
For a few Parameter Set values, all Parameter Set storage locations are initialized, after the sequence start, with:

- frame_cropping_rect_right_offset = 0
- frame_cropping_rect_top_offset = 0
- frame_cropping_rect_bottom_offset = 0
- Aspect_ratio_info = 0 (undefined)
- Video_signal_type = 101b (unspecified)

Recommended for adoption. Covered in high-level syntax output.

- When are ParSet values allowed to change
  - Some at sequence, some at GOP, some at Picture level

Parameter	Change Category	Remark
Log2_max_frame_num_minus_4	IDR	
num_of_reference_pictures	IDR	to facilitate resource allocation
required_frame_num_update_behaviour	IDR	
frame_width_in_MBs_minus1	IDR	
frame_height_in_MBs_minus1	IDR	
frame_cropping_rect_*_offset	Picture	to enable trick modes
aspect_ratio_info & parameters	IDR	to allow camera changes
video_signal_type & parameters	IDR	to allow camera changes
entropy_coding_mode	Picture	Author has no preference
motion_resolution	Picture	no preference
constrained_intra_prediction_flag	IDR	as per Porto Seguro
timing_info & parameters	IDR	to allow camera changes, MCU
Num_slice_groups & parameters	Picture	to tune error resilience strength

Recommended for adoption. Covered in high-level syntax output.

Convey Profile & Level in parameter set. Agreed. Covered in high-level syntax output.

- Limitation of Maximum Number of simultaneous Parameter Sets

Recommended for adoption. Clarify … Covered in other notes.

4.15.6. Picture Layer

JVT-D066 P2.0/3.1 [Wenger+] Removal of the picture layer

Proposed is the removal of the Picture Layer (introduced in Fairfax) and move its contents either to the Slice Header or to the Parameter Sets. This is reported to lead to a very low additional overhead of 0.1% in bit rate for error free environments, and to save 1% or more (depending on the use of start codes) on error free environments.

Summary: The picture header is evil. Move some of its syntax to a parameter set level and some to the slice header level. Causes conflict between syntax for error resilient environments and for error-free environments. Harms architectural design. Picture headers tend to grow over time.

Put MMCO only in slice header.

Editorial remark: replace start_mb_address with start_mb_x and start_mb_y.

Covered in high-level syntax output.

JVT-D087 P2.0/3.1 [Walker+] Network Adaptation Layer and High-Level Syntax
" This contribution presents several modifications intended to simplify and cleanup the NAL design: replacing the picture header with slice-level parameter sets, some modifications to slice header to move some information to slice parameters, as well as a reduction in the number of NAL unit types. It also presents some constraints on the ordering of NAL unit in an AVC stream."

Similar to D066 regarding picture header issue.

Advocates aggressively moving things into dynamic parameter set from current slice and picture header.

Allow multiple parameter sets to be sent in on NAL unit

Simplify the NAL types. No mixed picture versus non-mixed picture indication.

IDR indication.

User data, various SEI-related issues.

Ordering of NAL units

Outcome is reflected in high-level syntax output document.

JVT-D094 P2.0/3.1 [Hannuksela+] Modifications to High-Level Syntax and Semantics
"The contribution proposes replacement of the picture header with the picture parameter set. Furthermore, the contribution proposes an addition to the parameter set, a simplification to the definition of the first byte of the NAL unit syntax, and straightforward changes to the slice header syntax. It also presents a high-level architecture description of the JVT codec."

Similar to D066 regarding picture header issue but contains issues regarding:
- Parameter Sets & Picture Header
- NAL Unit Types
- SEI Messages
- Constraints on Order of NAL Units

Covered in high-level syntax output.

4.15.7. High-Level Activity Review Tues Morning

Two sets of parameters: I-GOP parameter set (max of 16 of these) and picture parameter set (max of 64 of these)
Slice header refers to picture parameter set: all slices of a picture refer to the same parameter set
Definition of when parameter set take effect
Parameter set values can be relative to default values (no update relative to remembered values)

Note: Decoding process depends on frame/field, not interlace/progressive -- fix this if different in the spec.

Decoding order = picture delivery order

ASO open issue.

Output document to capture the result.

4.15.8. NAL Unit Order

JVT-D093 P2.2/3.1 [Hannuksela] On NAL Unit Order
"We propose unified terminology for decoding, transmission, and display order of data units. Restrictions and liberties on NAL unit transmission and decoding order are presented. It is proposed the transmission order and decoding order of NAL units may not be the same, and the NAL decoder operation for rearranging the NAL units in decoding order for the VCL decoder is presented."

Coded order, coding order, decoding order, data dependency order, bitstream order, transmission order – lots of terms – what do we mean? Need consistent, defined terms.

What are order constraints? Does the network need to convey an order to the decoder.

ASO (within a picture) is another issue – other docs discuss that.

Parameter set information shall be received before any NAL unit that references it.

SEI shall be associated with the next transmitted slice or data partition.

Examples of possibility of decoding order differing from transmission order:

Example: Pre-delivery of I-frame for commercial to be shown later. Allow reception of pictures and leaving them in a pre-decoding bitstream buffer until decoding them later (keep them in the input bit buffer). Perhaps not know the presentation time when a picture is sent.

Example: Gateway reordering example.

Example: Retransmission to ensure anchor picture reception (storing in pre-decoding bitstream buffer to deal with transmission-layer jitter/loss).

Some of these issues could be handled by buffering in the transport layer.

Concept of sub-sequence identifier: NAL decoder shall give the VCL decoder pictures of increasing picture number with the same sub-sequence identifier. (A GOP being approximately the same thing as a sub-sequence.)

This is pre-decoding bitstream buffering. Remark: Could serve these applications with post-decoder picture buffering.

Conclusion: slices from different pictures will not be mixed. See high-level syntax output

4.15.9. HRD

JVT-D131 P2.2.1 [Viscito] HRD and Related Issues

Modifications for variable frame timing, multiple CBR cases,

Open issues identified by author: Post-decoder buffer issue, 90 kHz & representation of initial decoder buffer removal delay, semantic constraints on clock tick value may be needed.

Comment: How would you know the removal time? An expression of the buffer removal time for each picture is expressed in syntax.

HRD defines compliance from the content of the stream alone. Places constraints on the size of coded pictures and the timing of their life in a hypothetical pre-decoding buffer. Can serve as a timing model for a real decoder or system-level mux.

Two levels of constraint.

VCL content includes slice_header() and slice_data(). Non-VCL content includes SEI NAL units, GOP parameter set NAL units, picture parameter set NAL units.

Are Annex B zero_byte and one_byte syntax elements inside or outside of the NAL? Outside.

What is the VCL? What is in it? There is a VCL concept in the draft but no use of that term to distinguish between some syntax. Action item: Define the VCL syntax to consist of the slice_header() and the slice_data().

Either the HRD applies to the NAL units that contain slice_layer_no_partitioning(), dpa_layer_rbsp(), dpb_layer_rbsp(), dpc_layer_rbsp(), filler_data_rbsp(), or it applies to the "TEL" (transport encapsulation layer). One such TEL is the Byte Stream format. Others are defined externally. Those external specifications must specify what is counted as TEL data in addition to the NAL units carried in that TEL.

Time is needed for the HRD operation. Data flows into the pre-decoder buffer – at what rate? pictures are removed from it – when? pictures are removed from the post-decoder buffer – when? This information is necessary to HRD operation.

Difference in starting point in time between two random access points can cause a small difference in temporal location of picture removal times – that difference shall not cause buffer constraint violation.

Low delay and delay tolerant scenarios. Need for more than one big picture in a row? Yes. Change that part of the original proposal.

Spacing between decoding times must be greater than the minimum picture interval specified in the level to which the bitstream complies

Need for low delay indicator bit?

Constant bit rate vs. variable bit rate flag.

Pre-decoder removal delay.

1.Need to define precisely what is in the "VCL" and what is in the "NAL"
2.Reintroduce the low-delay flag. Remove the restriction that the picture after the big picture must meet the underflow constraint. Big pictures are disallowed in delay-tolerant.
3.Add statement to the Removal Time Consistency constraint specifying that all pre-decoder constraints must be satisfied regardless of the Buffering Period chosen as the starting point for compliance checking.
4.Add a restriction that the minimum interval between consecutive pre-decoder removal times (i.e. the minimum pre-decoder removal delay) must be no smaller than the minimum picture interval (which BTW is not normatively defined – inverse of max frame rate).
5.Add an explicit statement about overflow prevention.

Recommendations:
Bug fix: Max Frame Rate is not normatively defined as intended in May meeting. It is specified in the non-normative part. Move the max frame rate limit number from the non-normative part to normative part.

Group recommends adoption of JVT-D131, subject to review before next meeting.

Covered in HRD output doc.

Set up an Ad-hoc group with the following mandate:

AHG: (chair: Viscito) Mandate – review impact of HRD on real system operations; look at the restriction on the minimum removal delay with the aim to relax it, if useful. Define post-decoder buffer for inclusion in HRD. Discuss the merging of the PDB (Post Decoder Buffer) with the Reference Memory

JVT-D112 P2.2 [Lim+] Default Leaky Bucket for HRD
"The current HRD design requires the peak transmission rate R or buffer size B to select a suitable leaky bucket parameters from a set of leaky bucket parameters. This document proposes to select a "default" leaky bucket parameters from a set of leaky bucket parameters for cases when peak transmission rate R or buffer size B cannot be determined. The "default" leaky bucket parameters is based on MPEG-2's VBV model for CBR and VBR modes"

Advocates having syntax to tell the decoder what is its incoming bit rate.

May be an issue outside the scope of the video standard?

Recommendation: Text in CD to be clarified to make sure that definition of Max Rate and Max buffer size is clearly understood with regard to leaky buffer model.

Text should explicitly state that bounding box defined by (R,B) (0,0) must contain at least 1 leaky bucket.

Covered in HRD output doc.

JVT-D111 P2.2 [Lim+] Post Decoder Buffer for HRD
"The current HRD does not describe the post decoder buffer required for the reordering of B pictures. The objectives of this proposal are to address the problem of post decoder buffer for sequences with B pictures and to provide possible solutions to solve the problem."

Question: Is this issue within the scope of our standard?

Ultimate issue is the combination of decoding and display memory capacity.

One solution: Constrain the number of temporally future. Just a level restriction with no syntax.

Second solution: share memory between reference picture and post-decoder buffer.

Covered in HRD output doc.

JVT-D125 P2.2.1/3.1 [Chen] An Issue with Regard to B-picture Coding
"This document points out a problem on the picture delay in H.264 B-picture coding and proposes a way to fix it."

See high-level syntax output & HRD output.

JVT-D139 ----- [-----] Withdrawn

Recommendations on picture re-ordering subject area: Post decoder display buffer concept is accepted in principle. Needs to be harmonized with JVT-D017d2 and see if it is not already there. Delay in the display of the first decoded picture in a sequence needs to be signaled in the parameter set. Delay is in terms of number of fames (for interlaced video the delay in terms of number fields is 2x number of frames). Need to make sure that it works with variable frame rate.

Max Virtual post decoder buffer size to be specified in units of Byte / time etc (needs to ne discussed further).

Pictures are put in the PDB at the decoding time and are taken out of PDB at the presentation/display time.

Covered in HRD output doc. Further study needed to finalize.

JVT-D075 P2.2/3.1 [Kimata] A Proposal on B-picture Related Issue

"In CD, the reference picture for backward prediction is selectable. When more than one future reference picture can be selected, the system has to be capable of storing multiple output pictures before display. This document proposes the new high level syntax elements to indicate the number of frames to be delayed to output and it proposes the method of operating output picture memory. The proposed method enable the decoder to know the presentation order without timing information."

Propose explicit definition of output picture memory

Syntax for number of future pictures and number of "delta delay" pictures

Group conversation on these three contributions:

Comment: Profiling issues, is a matter of defining amount of memory needed for decode and display. Can define size of post-decoder buffering and ensure no production of dependency?

Comment: How about PTS=DTS for any non-stored picture?

How to constrain? How about setting PTS=DTS for some (any one) picture in the buffer?

Could just say your only responsibility is to decode, not display and not deal with the problem.

Do we care about this issue? Most people seem to care.

Output of decoder must be in display order?

Limit the total combined memory used for reference frame memory and post-decoder memory. Specify a limit on this in bytes and a limit on allowed number of stored frames.

Example: Increase current limits by 1 to have a place to put the decoded picture.

Covered in HRD output doc.

4.16. Relation to Systems

4.16.1. Relation to File Format

JVT-D092 P2.2.1/3.1 [Hannuksela] The Future of the Interim File Format
The contribution proposes taking the Interim File Format defined in Annex C of the JVT Working Draft (JVT-C039) as an integral part of the JVT coding standard.

Requirement to have file format with support of specific features. These things are not fulfilled in the current JVT CD. Intent has been to design file format as MP4 or ISO Media File format. These are based on a "box" structure design. Sony (Singer) document from Jeju Island cited as a reference on this topic. Proponent indicates features of the prior Annex C design that need support:

- Support of access unit fragmentation (access to slices)
- Multiple streams in one file with switching

- Disposable sub-sequences of pictures (a temporal scalability feature).
- Support of parameter set handling.

Proposal to do this on the current JVT (phase 1) schedule. Comment: Even if we wanted a JVT-codec-only file format, is that a realistic suggestion?

Discuss with MPEG Systems – hope for committee draft in Klagenfurt covering at least items 1 and 3. Addressed by the MPEG parent body with significant progress at its meetings.

JVT-D102 Info. [Toma+] Comments on JVT-C143, File Format for JVT based on MP4
"Regarding the JVT storage in MP4, we propose the followings as respective contributions to MPEG.
1, Editorial and technical changes for the current spec of JVT stream encapsulation onto MP4.
2. Optimization to reduce the overhead.
3. Support for JVT stream in the fragmented MP4 file."

No discussion requested by proponent.
JVT-D103 ----- [-----] Withdrawn
JVT-D104 ----- [-----] Withdrawn

4.16.2. Relation to MPEG-2 Systems

JVT-D107 P2.2/3.1 [Matsui] Comments on ISO/IEC 13818-1:2000/AMD3 (JVT on MPEG-2)
"It is proposed to add constraints when JVT stream is encapsulated onto MPEG-2 TS and PS.
1) The use of the error_flag of NALU is limited in order to avoid duplication of the start codes and functionality.
2) For each picture, an SEI NALU that contains TR shall precede all other NALUs for a picture so that the receiver can calculate the presentation timestamp for each picture."

Resolved by specification at systems layer.

4.16.3. Relation to General Systems

JVT-D084 Info. [Singer+] Interface between Systems Layer and JVT Codec
"In this contribution we discuss the interface between the Systems Layer and the JVT CODEC in terms of the flow of information between the Systems Layer and the decoder including parameter sets, timing information, and "control" SEI messages."

Decoder principle A (JVT-D084 + Fairfax intent): reconstructed sample values are independent of TR subject to resolving the caused problems
Decoder principle B (alternative + previous to Fairfax): reconstructed sample values are independent of PTS and DTS but can be dependent on TR. Normally PTS and DTS are derived from TR. For Trick modes etc. PTS and DTS are maybe changed but TR shall never be changed.

If the lack of the TR does not affect performance of B picture coding, the concept of using TR for decoding sample values (as of WD-2) is abandoned. Otherwise TR will be reconsidered for decoding sample values.

General Principles:
- decoder can be setup using out-of-band parameter sets

- buffer management (HRD) is optional ? Not requiring HRD conformance must be part of a particular profile. Make this a subject of profile session if it is requested there. Define compliance within a certain environment ?

Are parameter sets buffered?: part of HRD discussion.

This contribution is basically a description of Fairfax intent and a discussion of related issues particularly including the handling of timing information. No new technical content.

Considered in high-level syntax activity & discussion of other related topics. Future activity planned to finalize this.

4.17. Profiles & Levels & Limits

4.17.1. Summary & Discussion: Joint session notes

Changes requested to affect profiling with respect to

- Dither
- ABP (add it to Main)
- ABT (make sure it works) [later confirmed]
- Error Resilience Features:
  - FMO/ASO
  - Data Partitioning
  - SP/SI

Possible compromises: Turn off deblocking at slice boundaries. Add SP/SI. Put ABT only at higher levels. Add ABP. Limit of 4 slice groups.

Look at application domains.
Try not to let the constraints get too ugly and become impossible to explain.
Need to show that a tool is a real burden before saying it needs to be removed from a profile.

ISG: If software available, can do an analysis. Try to formulate a problem question in a way that can lead to an analysis. Parallelization can be part of the analysis.

Get timetable for software availability.

Deblocking and random order.

Levels

- JVT-D067 changing the levels model
- JVT-D076 error in table
- JVT-D091
- JVT-D131 Max bit rate and HRD size (note that there are also relevant contributions to Fairfax)
- JVT-D064
- USNB: Should there be a level 5?

4.17.2. Complexity

JVT-D138/M8547 Info. [Bormans+] Complexity Analysis of the AVC Codec

Information document. Appears to contain valuable information and members are encouraged to study this.

JVT-D153/M8696 Info [Ravasi+] A Computational Complexity Comparison of MPEG-4 and JVT Codecs

Information document. Appears to contain valuable information and members are encouraged to study this.

4.17.3. Profiles

JVT-D133 Prop(removal) [Syed+] Error Resilience Tools Belong in Error Resilience Profiles
This contribution proposes that error resilience tools including FMO, ASO, Data Partitioning, SP& SI switching and others that are in the committee draft should be put into an error resilience profile. Advocates that these tools should be removed from baseline and Main profiles if existing within these profiles unless they do not add significant costs to the decoders.

Contribution from Broadcom, LSI Logic, Motorola, Scientific-Atlanta, and CableLabs (on behalf of its members).

Addressed by removal of FMO/ASO from Main profile.

JVT-D085 Prop.(Profile-only) [Walker+] Proposal for a JVT Streaming Profile
This contribution proposes a new Streaming Profile for JVT. The proposed profile is targeted at applications delivering streaming video over IP networks, including both wired and wireless networks.

Contribution from Sony, Nokia, RealNetworks, Texas Instruments, Siemens, BTexact, Philips Research, and Motorola.

Addressed by addition of new Profile X.

4.17.4. FMO & Profiles

JVT-D121 Prop.(Profile tool removal) [Syed+] FMO Cost Burden to Decoders Operating in Reliable Nets
This contribution proposes that FMO be removed from the baseline and main profile and indicates some cost burdens that support this position.

Contribution from Broadcom, LSI Logic, Scientific- Atlanta, Motorola, and CableLabs (on behalf of its members).

Addressed by removal of FMO/ASO from Main profile.

JVT-D115 P2.0/3.1 [Mazzarotto] ASO & FMO Impact on AVC Compliance and Complexity

This document analyzes the impact of the AVC Arbitrary Slice Order (ASO) and Flexible Macroblock Order (FMO) tools on AVC visual bitstreams' compliance testing, and on AVC decoders' implementation complexity. Based on this analysis, it proposes to: a) limit ASO within pictures, b) create a new profile that supports both these tools, and, c) relieve other decoders from supporting ASO and FMO by removing these two tools from the Baseline and Main profiles.

Contribution from Broadcom, LSI Logic, Scientific- Atlanta, Motorola, and CableLabs (on behalf of its members).

Addressed by removal of FMO/ASO from Main profile.

JVT-D063 Info. [Wenger+] FMO-101
This document provides an introduction to FMO, and discusses in detail decoder implementation aspects of FMO, in particular scan-order processing of macroblocks when FMO is in use.

Addressed by FMO support in Baseline and Profile X.

4.17.5. Levels & Limits

Wednesday

Max pic parameter sets: 64
Max gop parameter sets: 16

Put profile & level indicator in gop parameter set. Yes.

JVT-D076 P2.0/3.1(Ed.) [Lindbergh] Editorial changes to JVT draft
"Proposes minor editorial corrections to Annex A of the JVT CD
text. The maximum picture size for Level 4 should be 8192 macroblocks as
agreed in the Fairfax meeting, not 9660."

Adopted.

Note 172 frame rate limit editorial problem – normative requirement in non-normative text. Fix that.

Vertical MV range:
[-128, 127] level 2 and below,
[-256, 255] level 3 and below

HRD Buffer sizes cluster around 1 second (double if streaming).

4.17.6. Further Study Topic: Unspecified Limits

JVT-D091 P2.0/3.1. [Hannuksela] On Level Definitions