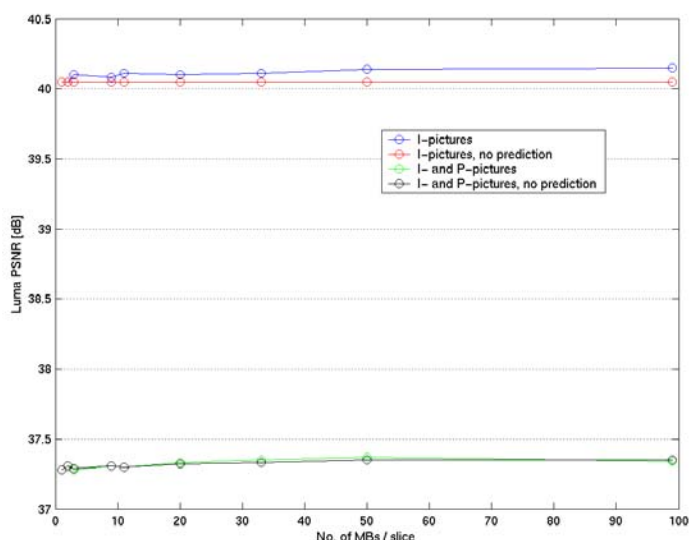


Some of the parameters of the here presented examples are given in the following:

- Sequence: Foreman
- Size: QCIF
- Sequence type: IPPP...
- No. of frames: 20
- QP: 12/15
- Skip: 2
- MV resolution: 1/4-pel
- ME search range: 16
- Reference frames: 1
- CW mode: UVLC
- File type: bit stream
- Partition mode: no DP

Error treatment in JM-1.9

So far, H.26L offers powerful error handling capabilities. They are applied at slice level and meant to treat slice losses. In Annex D, the current Working Draft [WD-2r3] mentions non-normative error concealment methods like e.g. pixel interpolation by means of data from concurrent spatial areas (for INTRA MBs) and MVD estimation (for INTER MBs). However, one weakness is that JM-1.9 cannot treat residual errors. Rather, error affected slices are discarded/concealed as a whole, hereby also overriding not affected, i.e. valid data. In fact, a single bit error may lead to a complete slice be thrown away.



Motivation

If the desired type of communication is conversational, no retransmissions are applied at the RLP/RLC level, hereby leading to residual bit and burst errors. In order to not let these errors propagate, slices sizes have to be kept small. Slices are coded independently of each other, or in other words, there is neither spatial nor temporal prediction between two slices, even though they might be neighboring. Inhibited prediction leads to a negligibly degraded quality of the decoded image. This is depicted in the last Figure for the parameters mentioned above. Since prediction becomes worse, the energy of the prediction error grows as more and more slices are in a picture, hereby accompanied by a significant increase in bandwidth, as depicted in the next Figure. The growing of the bit rate is moderate down to approximately 11 macroblocks (MBs) per slice (mbps). This is the starting point for the proposal.

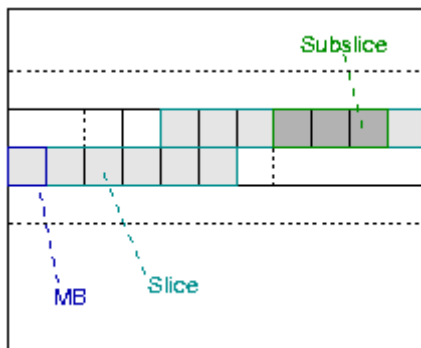
Proposal Part I: Bit rate reduction

The proposal is based on the observation that, for a large number of slices per picture (slppic) - which is the same as a small number of mbpsl - a lot of information is always repeated by the slice header. However, most of this information is the same for neighboring MBs and therefore superfluous to transmit many times.

Compared to the payload (i.e. slice data), the size of the slice header is relatively large, especially when dealing with P-pictures and low bit rates. The slice header size may vary from 39 to 162 bits with typical values being around 70-80 bits for short QCIF sequences. For the example above, this leads to the following overhead ratios which were already visualized in the first Figure.

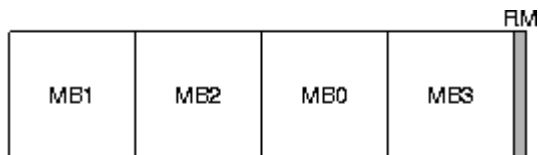
No. of MBs/slice	99	50	33	20	11	9	3	2	1
No. of slices/frame	1	2	3	5	9	11	33	50	99
Av. overhead rat.I (%)	.002	.0043	.0066	.0107	.019	.023	.0762	.1085	.1901
Av. overhead rat.P (%)	.0084	.0181	.0291	.0488	.0861	.1095	.3152	.4147	.5743

By further splitting one slice into several subslices (mini slices or Group of Blocks (GOBs), segments, whatever) containing MBs that all have the same slice information, the overhead under erroneous conditions can be significantly reduced. See the following Figure for an example how this could look like in the picture plane. Within one slice, there are several subslices. The number of MBs per subslice can be chosen by the encoder from slice to slice. This could be a preset value or, if additional information - like e.g. effective error rate of residual bit errors - is available by means of a feed-back channel, can be altered during encoding to adaptively reflect the potentially non-stationary channel conditions. The size of that number should also account for the value of the syntax element RUN, as many RUNs larger than zero, which can be caused by e.g. large QPs, let the use of RMs become inefficient.



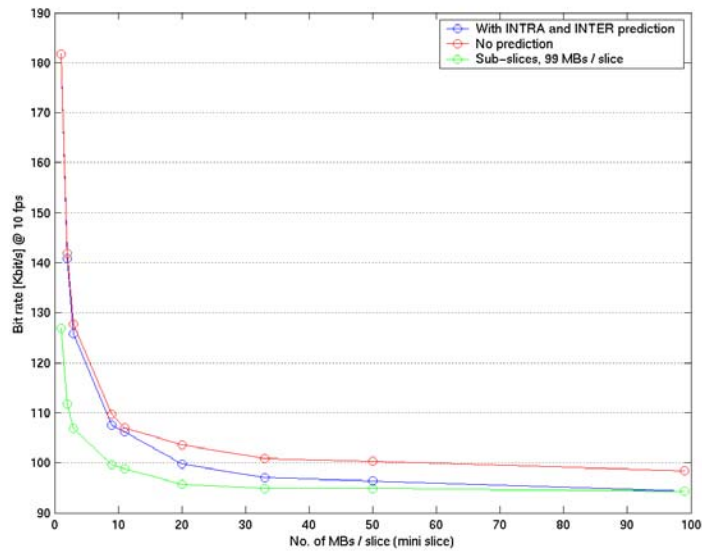
The number of MBs per subslice is signaled to the decoder by means of the SM. Using a variable-length (VL) code, the SM consists of a CW of length 31 [bits] and the info bits $(0)_2$. (The earlier proposal of having a picture start code, which is the same like the SM, except the info being $(1)_2$ instead of $(0)_2$, was discarded. Now, everything is done one a slice level.) However, in order to be able to split the received code stream again into slices, only the length of the SM is relevant for the decoder. This leaves the 15 info bits of the SM unused. It is hence proposed to utilize the info bits of the slice header to code the number of MBs of one subslice. Since it is possible to fill out one slice by a single subslice, the SM's info bits must provide for to convey a number of the size up to the maximum number of MBs per picture. The current Profile&Level specification of H26L [JVT-B108] names the maximum picture format to be 4096x3072, which means a maximum number of 16x16 MBs of 49152.

In order to represent this number by the SM's info bits, 16 bits would be necessary, which in turn gives a SM with length 33. In this contribution, however, 15 bits were used since the experiments were carried out for small picture sizes only.

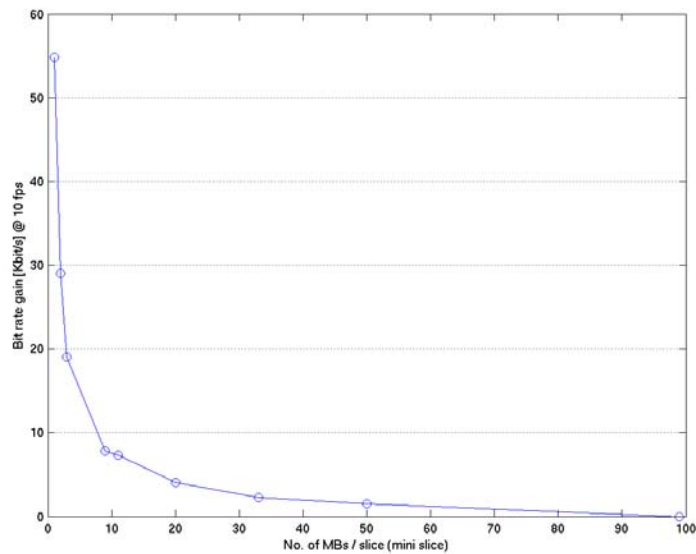


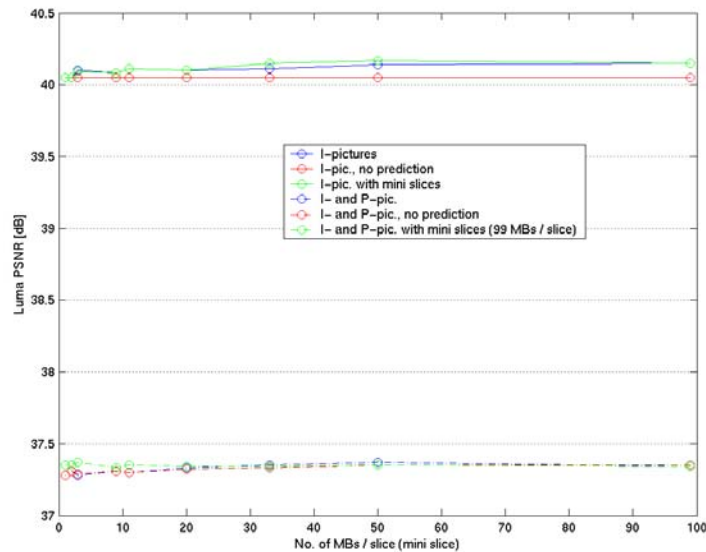
The structure of one subslice is sketched in the Figure above. The number of MBs per subslice is passed to the encoder as a parameter. In order to inhibit error propagation, no prediction is granted over subslice boundaries. The same has to be applied to the RUN parameter. Slice boundaries are hereby included. To prevent error propagation, a unique resynchronization markers (RM), that cannot be emulated by the coding engine, is inserted in the code stream after every subslice. By doing so, the same effect of a slice marker (SM) is achieved, but without the large amount of overhead ordinarily associated with such markers. A RM can be written arbitrarily, i.e. bit-aligned. It is proposed for complexity reasons to take a VL code word as RM, e.g. the CW defined by length 29 and info bits 16383. However, then today's used VL code has to be made finite, i.e. the code table has to be made finite, such that it is not possible for the coding engine to emulate the RM by length and info part. This limitation has to be done for implementation purposes anyway. No RM is placed directly before a SM. The number of MBs

per subslice is constant within the subslice with the exception of the last subslice which may be smaller. Subslices do not stretch over slice boundaries.



This technique leads to a considerably reduced overhead, as shown in the last Figure. The gain hereby accomplished is pictured in the following Plot. The Figure after that is probative for that the quality of the decoded pictures does not suffer under error-free transmission. In addition, use of RMs limits potential error propagation over subslice boundaries and allows hereby to localize errors better.





Proposal Part II: Bit error handling

Provided information about the reliability of the code stream is available, the decoder might skip, copy or conceal the MBs contained in one subslice while not affecting decoding of MBs in subslices assumed reliable, i.e. error-free. However, it cannot be expected that this information is available for every application. Therefore, in case of having to face residual errors, steps have to be taken to enable the decoder to detect these errors by own initiative.

Of major help to increase robustness is a new code table, VLCD, which was already introduced in [JVT-B034r1]. The name refers to the variable-length code (VLC) which is described in H.263 Annex D. VLCD has the same coding efficiency as UVLC (both in interlaced or non-interlaced form), which is tailored for low bit error rates. Additionally, it is a reversible variable-length code (a property which is not exploited here), and as explained in [JVT-B034r1], it does not resynchronize after single bit errors. This is used here for error detection purposes. In order to work, the nonresync property of VLCD has to be combined with data partitioning, provided by the RMs. A sufficiently high number of subslices per slice is hereby assumed, which corresponds to a moderate bit error rate, which in turn is necessary to assure synchronization is not regained under decoding,

Practically, the code stream is scanned for RMs before decoding. Between two RMS there must be a subslice. Each of these code segments is then decoded on a trial basis to find out if sync is lost. Sync is lost when a RM is reached and decoding of the current symbol is not finished yet. Not contaminated subslices are further passed to the decoder, contaminated ones are just skipped, and the number of skipped MBs is added to the next occurring RUN. This requires that the decoder does not skip RUN for INTRA blocks (documented correctly in WD2r3 but implemented differently in JM-1.9). Thus, the decoder automatically conceals subslices without increased complexity.

Results

For this kind of experiments, the common testing conditions [VCEG-N80] do not make sense. The expected gain of error concealment methods is mainly of visual nature, so PSNR is less important in these matters. From the conditions, one set of parameters is picked: Foreman, QCIF, bit error patterns 3 and 4 (BER $5.1e-4$ and $1.7e-4$, respectively), 7.5 fps, various starting positions of the pattern files. In visual comparisons, the proposed method has to compete with the reference system, which makes uses of the same parameters. The reference codec uses

non-interlaced UVLC and error concealment as specified in Annex D of WD2r3. The number of MBs per slice and MBs per subslice, both in the new system, have been chosen such that the transmitted bit rate is approximately equal for both systems.

The following example images are picked from this evaluation. They show the same decoded picture of a sequence for identical error patterns and starting position; the reference system on the left, and the proposed system on the right. The left system splits one picture into five slides (of three times 20 and once 19 MBs); the right system uses two slices only and additionally 9 MBs per subslice. The bit rate is ca. 108 Kbit/s. This demonstrates clearly the superiority of the proposal made here. More, the results are consistent for all testing conditions.

The reference codec has several breakdowns under execution, pointing to some bugs still hiding in the TML. However, there are limitations for both systems with regard to termination of decoding when e.g. a slice header is hit or when sync is re-established after several bit/burst errors. A possible remedy here in the first case is repetition of the header as in MPEG-4 (header extension code, HEC).



Conclusions

It has been shown that it pays to maintain valid data during decoding. Residual bit errors could be successfully located roughly, and their impact has been limited significantly. At the same time, complexity was reduced since the proposed system does not employ the same error concealment like the reference system.

Two minor drawbacks remain: VLCD is slightly more complex than today's non-interleaved UVLC code. And second, in case info bits are hit by errors, they are not recognized as such. This is valid for both UVLC and VLCD.

The proposal is summarized in the following.

Summerized proposal

Normative part (overhead reduction)

- Variable-size subslice structure with limited prediction
- RMs between subslice boundaries (length 29, info bits 16383)
- Finite VL code
- Utilization of SM's info bits
- Enlargement of SM to 33 bits
- MB skip by RUN also for INTRA blocks

Non-normative part (bit error robustness)

- Use of VLCD for error detection (gives RM $(011...110)_2$)
- Optionally HEC

Outlook

Even though the proposed techniques have been proven to achieve their goals efficiently, some further aspects remain. The concept of subslices, data partitioning and RMs could also be applied to the CABAC or other entropy code (e.g. CVLC), provided that sync is lost after error occurrence. At the end of one code segment, a certain short bit pattern, e.g. $(1010)_2$ is arithmetically encoded (without resetting the contexts), and the resulting bits are appended to the code stream. If this pattern is not computed again by the arithmetic decoding engine, an error can be flagged. This means additionally, in contrast to VLCD where the marker has to be placed MB-wise, that the marker here can be written equally spaced (equal number of bits in between) such that regions with high motion (i.e.. with a large number of bits for one MBs) are especially well protected.

Further experiments should verify the assumed gain of application of the proposed header extension code. Also, the proposed system uses only the bit stream format of the compressed stream. The partition approach has not been investigated and therefore remains.

References

[N80] V. Varsa, M. Karczewicz, G. Roth, R.Sjöberg, T. Stockhammer, and G. Liebl. Common Test Conditions for RTP/IP over 3GPP/3GPP2. VCEG-N80, December 2001

[WD-2r3] T. Wiegand and G. Sullivan. Working Draft Number 2, Revision 3. JVT-B118r3, March 2002

[JVT-B108] D. Lindbergh. Profiles&Levels. JVT-B108, March 2002

[JVT-B034r1] T. Halbach. Enhanced variable-length coding. JVT-B034, January 2002

(Append for Proposal Documents)

JVT Patent Disclosure Form

International Telecommunication Union
Telecommunication Standardization Sector



International Organization for Standardization



International Electrotechnical Commission



Joint Video Coding Experts Group - *Patent Disclosure Form*

(Typically one per contribution and one per Standard | Recommendation)

Please send to:

JVT Rapporteur Gary Sullivan, Microsoft Corp., One Microsoft Way, Bldg. 9, Redmond WA 98052-6399, USA
Email (preferred): Gary.Sullivan@itu.int Fax: +1 425 706 7329 (+1 425 70MSFAX)

This form provides the ITU-T | ISO/IEC Joint Video Coding Experts Group (JVT) with information about the patent status of techniques used in or proposed for incorporation in a Recommendation | Standard. JVT requires that all technical contributions be accompanied with this form. *Anyone* with knowledge of any patent affecting the use of JVT work, of their own or of any other entity (“third parties”), is strongly encouraged to submit this form as well.

This information will be maintained in a “living list” by JVT during the progress of their work, on a best effort basis. If a given technical proposal is not incorporated in a Recommendation | Standard, the relevant patent information will be removed from the “living list”. The intent is that the JVT experts should know in advance of any patent issues with particular proposals or techniques, so that these may be addressed well before final approval.

This is not a binding legal document; it is provided to JVT for information only, on a best effort, good faith basis. Please submit corrected or updated forms if your knowledge or situation changes.

This form is *not* a substitute for the *ITU ISO IEC Patent Statement and Licensing Declaration*, which should be submitted by Patent Holders to the ITU TSB Director and ISO Secretary General before final approval.

<u>Submitting Organization or Person:</u>	
Organization name	_____
Mailing address	_____
Country	_____
Contact person	_____
Telephone	_____
Fax	_____
Email	_____
Place and date of submission	_____
<u>Relevant Recommendation Standard and, if applicable, Contribution:</u>	
Name (ex: “JVT”)	_____
Title	_____
Contribution number	_____

(Form continues on next page)

Disclosure information – Submitting Organization/Person (choose one box)



2.0 The submitter is not aware of having any granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

or,

The submitter (Patent Holder) has granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution. In which case,



2.1 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a free license to an unrestricted number of applicants on a worldwide, non-discriminatory basis to manufacture, use and/or sell implementations of the above Recommendation | Standard.



2.2 The Patent Holder is prepared to grant – on the basis of reciprocity for the above Recommendation | Standard – a license to an unrestricted number of applicants on a worldwide, non-discriminatory basis and on reasonable terms and conditions to manufacture, use and/ or sell implementations of the above Recommendation | Standard.

Such negotiations are left to the parties concerned and are performed outside the ITU | ISO/IEC.



2.2.1 The same as box 2.2 above, but in addition the Patent Holder is prepared to grant a “royalty-free” license to anyone on condition that all other patent holders do the same.



2.3 The Patent Holder is unwilling to grant licenses according to the provisions of either 2.1, 2.2, or 2.2.1 above. In this case, the following information must be provided as part of this declaration:

- patent registration/application number;
- an indication of which portions of the Recommendation | Standard are affected.
- a description of the patent claims covering the Recommendation | Standard;

*In the case of any box **other than 2.0** above, please provide the following:*

Patent number(s)/status _____

Inventor(s)/Assignee(s) _____

Relevance to JVT _____

Any other remarks: _____

(please provide attachments if more space is needed)

(form continues on next page)

Third party patent information – fill in based on your best knowledge of relevant patents granted, pending, or planned by other people or by organizations other than your own.

Disclosure information – Third Party Patents (choose one box)



3.1 The submitter is not aware of any granted, pending, or planned patents *held by third parties* associated with the technical content of the Recommendation | Standard or Contribution.



3.2 The submitter believes third parties may have granted, pending, or planned patents associated with the technical content of the Recommendation | Standard or Contribution.

For box 3.2, please provide as much information as is known (provide attachments if more space needed) - JVT will attempt to contact third parties to obtain more information:

3rd party name(s) _____

Mailing address _____

Country _____

Contact person _____

Telephone _____

Fax _____

Email _____

Patent number/status _____

Inventor/Assignee _____

Relevance to JVT _____

Any other comments or remarks: