Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
2nd Meeting: Geneva, CH, Jan. 29 - Feb. 1, 2002

Document: JVT-B011
Filename: JVT-B011.doc
Generated: 2002-01-25

Title: **AHG Report: Loop Filter**

Status: Input Document to JVT

Purpose: Report

Author(s) or Contact(s): Peter List
Deutsche Telekom
Am Kavalleriesand 3
D-64295 Darmstadt, Germany
Tel: +49 6151 83-3878
Email: Peter.List@t-systems.com

Source: Deutsche Telekom, T-Systems

The charter of the loop filter Ad-Hoc was:
"To investigate quality and complexity issues for loop filter design in the JVT codec and to assess the potential for improved visual quality, reduced decoder computational complexity, and enhanced design simplicity."
70 members are currently subscribed. Only a few of them participated actively. The group's communication was taking place on a separate reflector (jvt-loopfilter@berkom.de).

The main focus was on reducing filter complexity, since the filter was reported to consume more than 50% of the decoding time of an optimized decoder.

It was identified, that the biggest chunk of the loop filter complexity comes from copying due to the non recusiveness of the filter. Changing to recursive "in place filtering" did not significantly influence the quality and was basically agreed upon. The only objection was concerned with highly pipelined architectures, that might suffer from the large number of data dependencies introduced by in place filtering. This was resolved by being non recursive for INTRA strong filtering.

New code was provided, that contained some simplifications but basically has the same "spirit" and results:

- in "normal filtering" (in contrast to INTRA strong filtering) only the two edge pixels are filtered for chrominance
- Normal filtering is recursive now.
- calculation of the Strength parameter was changed as proposed in VCEG-N17
- calculation of n, ap, aq and the clipping boundary c0 was simplified
- a potential problem with differing QP-values of neighboring macroblocks was solved by averaging.

An independent verification said, the new code had twice the speed of the TML9.0 filter.
Taking silence as approval this simplified filter was accepted by the group to be the anchor for new proposals.

An other thread of the discussions dealt with the question whether a switch (on sequence level) should be provided, to be able to turn the filter off completely. This might be desirable for some high quality profiles were the filter would be of no use. An other solution would be to define these profiles to not use the filter at all. The bottom line of this is: do we require *all* profiles to be downward compatible with the baseline profile (which has the filter) or not. It lies certainly beyond the charter of this Ad Hoc to answer this question.

INTEL-1051
9,641,841

Again, the question of macro block based versus frame based was discussed in some detail. Most participants preferred the macro block based approach but it was stated, that for some architectures a frame based approach might be preferable. In principle both variations are possible for the filter as it is now. The only point missing, is to define that INTRA-prediction will be made on the *unfiltered* content of the reconstructed frame, which requires implementations that work on a macro block basis to store some unfiltered pixels for INTRA-prediction. The group agreed on this (except for one participant, who wanted to postpone the decision).

For the case, that the new filter proposal is adopted, some text for the JM2 document is annexed.

1.1 Deblocking Filter

After the reconstruction of a macroblock a conditional filtering of this macroblock takes place, that effects the *boundaries* of the 4x4 block structure. Filtering takes place on a macroblock level. In a first step the 16 pel of the 4 vertical edges (horizontal filtering) of the 4x4 raster are filtered. After that, the 4 horizontal edges (vertical filtering) follow. Note, that this process also affects the boundaries of the already reconstructed macroblocks above and to the right of the current macroblock. Frame edges are not filtered. Note, that intra prediction of the current macro block takes place on the *unfiltered* content of the already decoded neighbouring macroblocks. Depending on the implementation, these values have to be stored before filtering.

1.1.1 Content Dependant thresholds

For the purpose of filtering each 4x4 block edge in a reconstructed macroblock a filtering **Boundary Strength Bs** is assigned for luma:

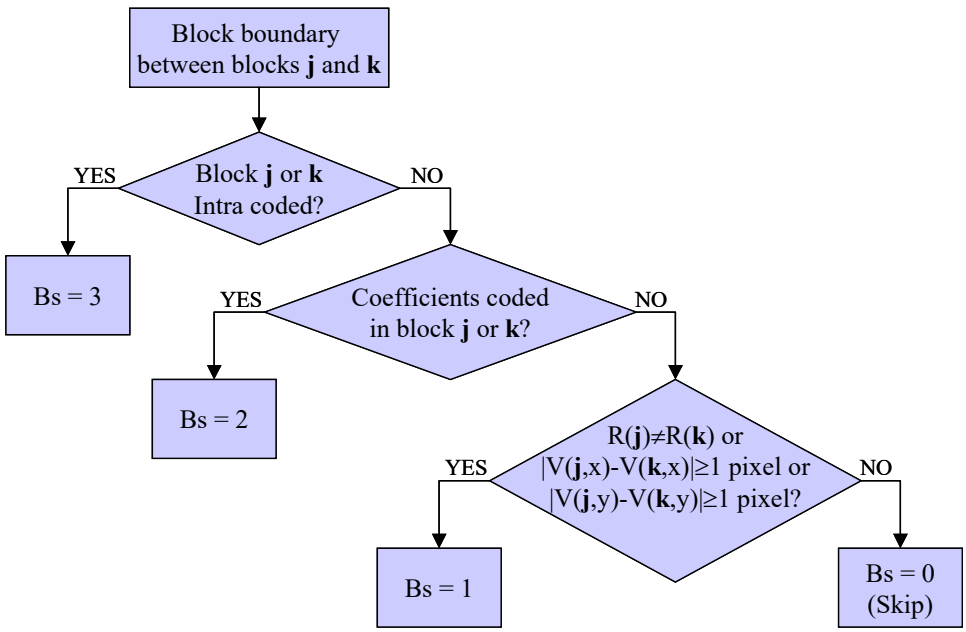

FIGURE 1

Flow chart for determining boundary strength (Bs), for the block boundary between two neighbouring blocks j and k.

Block boundaries of chroma blocks always correspond to a block boundary of luma blocks. Therefore the corresponding **Bs** for luma is also used for chroma boundaries.

The set of eight pixels across a 4x4 block horizontal or vertical boundary is denoted as

$$p_4,p_3,p_2,p_1 \mid q_1,q_2,q_3,q_4$$

with the actual boundary between p1 and q1. In the default mode up to two pixels can be updated as a result of the filtering process on both sides of the boundary (that is at most p_2, p_1, q_1, q_2). Filtering across a certain 4x4 block boundary is skipped all together if the corresponding **Bs** is equal to zero.

If Bs $\neq 0$ sets of pixels across this edge are only filtered if the condition

$$|p_1 - q_1| < \alpha \quad AND \quad |p_2 - p_1| < \beta \quad AND \quad |q_2 - q_1| < \beta$$

is true. The QP dependant thresholds α and β can be found in table 5.

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
α	3	3	3	3	3	3	3	3	3	3	4	4	5	6	8	10	13	17	21	26	30	35	43	48	55	64	77	96	128	128	192	192
β	0	0	0	0	0	0	0	0	4	4	4	5	5	5	7	7	7	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15

TABLE 1

QP dependent threshold parameters α and β.

1.1.2 The Filtering Process

Two sorts of filter are defined. In the default case the following filter will be used for p_1 and q_1

$$\Delta = Clip(-C,\ C,\ ((q_1 - p_1) << 2 + (p_2 - q_2) + 4) >> 3)$$
$$P_1 = Clip(0,\ 255,\ (p_1+\Delta))$$
$$Q_1 = Clip(0,\ 255,\ (q_1-\Delta))$$

The two intermediate threshold variables

$$a_p = |p3 - p1|$$
$$a_q = |q3 - q1|$$

are used to decide whether p_2 and q_2 are filtered. These pixels are only processed for luminance.
If (Luma && $a_p < \beta$) p_2 is filtered by:

$$P_2 = p_2 + Clip(-C0,\ C0,\ (p_3+P_1 - p_2<<1) >> 1)$$

If (Luma && $a_q < \beta$) q_2 is filtered by:

$$Q_2 = q_2 + Clip(-C0,\ C0,\ (q_3+Q_1 - q_2<<1) >> 1)$$

Where Clip() denotes a clipping function with the parameters Clip(Min, Max, Value)
and the clipping constants:

$$C0 = ClipTbl(QP,\ \mathbf{Bs})$$
$$C = C0 + a_p + a_q$$

ClipTbl (QP, **Bs**):

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Clip(qp,1)	0	0	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	4	4
Clip(qp,2)	0	0	0	0	0	0	0	0	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	6	6
Clip(qp,3)	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	3	3	3	3	4	4	4	5	5	6	6	6	8	8	9	10

TABLE 2

QP and Bs dependent Clipping thresholds

1.1.3 Stronger filtering for intra coded macroblocks

In case of intra coding and areas in the picture with little texture the subjective quality can be improved by stronger filtering across macroblock boundaries (that is: *not* 4x4 block boundaries). This stronger filtering is performed if:

$$\mathbf{Bs} == 3$$

and

$$a_p < \beta \text{ and } a_q < \beta$$

and

$$1 < |p_1-q_1| < QP / 4$$

In this case filtering is performed in the with the equations shown below.

$$P_1 = (p_3 + 2*p_2 + 2*p_1 + 2*q_1 + q_2 + 4) / 8$$
$$P_2 = (p_4 + 2*p_3 + 2*p_2 + 2*p_1 + q_1 + 4) / 8$$
$$Q_1 = (p_2 + 2*p_1 + 2*q_1 + 2*q_2 + q_3 + 4) / 8$$
$$Q_2 = (p_1 + 2*q_1 + 2*q_2 + 2*q_3 + q_4 + 4) / 8$$

Only for luma p3 and q3 are filtered:

$$P_3 = (2*p_4 + 3*p_3 + 2*p_2 + p_1 + 4) / 8$$
$$Q_3 = (2*q_4 + 3*q_3 + 2*q_2 + q_1 + 4) / 8$$