Adaptive Deblocking Filter

Peter List, Anthony Joch, Jani Lainema, Gisle Bjøntegaard, and Marta Karczewicz

Abstract—This paper describes the adaptive deblocking filter used in the H.264/MPEG-4 AVC video coding standard. The filter performs simple operations to detect and analyze artifacts on coded block boundaries and attenuates those by applying a selected filter.

Index Terms—Block-based coding, video coding, video filtering, video signal processing.

I. INTRODUCTION

THERE are two building blocks within the architecture of the H.264/MPEG-4 AVC video coding standard [1] which can be a source of blocking artifacts. The most significant one is the block-based integer discrete cosine transforms (DCTs) in intra and inter frame prediction error coding. Coarse quantization of the transform coefficients can cause visually disturbing discontinuities at the block boundaries [2]–[4]. The second source of blocking artifacts is motion compensated prediction. Motion compensated blocks are generated by copying interpolated pixel data from different locations of possibly different reference frames. Since there is almost never a perfect fit for this data, discontinuities on the edges of the copied blocks of data typically arise. Additionally, in the copying process, existing edge discontinuities in reference frames are carried into the interior of the block to be compensated. Although the small 4×4 sample transform size used in H.264/MPEG-4 AVC somewhat reduces the problem, a deblocking filter is still an advantageous tool to maximize coding performance.

There are two main approaches in integrating deblocking filters into video codecs. Deblocking filters can be used either as post filters or loop filters. Post filters only operate on the display buffer outside of the coding loop, and thus are not normative in the standardization process. Because their use is optional, post-filters offer maximum freedom for decoder implementations. On the contrary, loop filters operate within the coding loop. That is, the filtered frames are used as reference frames for motion compensation of subsequent coded frames. This forces all standard conformant decoders to perform identical filtering in order to stay in synchronization with the encoder. Naturally, a decoder can still perform post filtering in addition to the loop filtering if found necessary in a specific application.

Performing the filtering inside the coding loop has several advantages over post filtering. Firstly, the requirement of a loop filter *guarantees* a certain level of quality. This is especially important in modern communications systems where decoders of several manufacturers are used to decode distributed video material. With a loop filter in the codec design, content providers can safely assume that their material is processed by proper deblocking filters, guaranteeing the quality level expected by the producer.

Secondly, there is no need for an extra frame buffer in the decoder. In the post-filtering approach, the frame is typically decoded into a reference frame buffer. An additional frame buffer may be needed to store the filtered frame to be passed to the display device. In the loop-filtering approach, however, filtering can be carried out macroblock-wise during the decoding process, and the filtered output stored directly to the reference frame buffers.

Thirdly, empirical tests have shown that loop filtering typically improves both objective and subjective quality of video streams with significant reduction in decoder complexity compared to post filtering [3], [5]. Quality improvements are mainly due to the fact that filtered reference frames offer higher quality prediction for motion compensation. Reductions in computational complexity can be achieved by taking into account the fact that the image area in past frames is already filtered, and thereby optimizing the filtering process accordingly. Fig. 1 shows the reconstructed frame in a loop-filter (left) and a post-filter (right) based TML 8.5[1] system before loop/post filtering. It can be seen that the main coding artifact in the loop-filter case is the blockiness on the 4×4 grid caused by prediction error coding. This artifact can be efficiently compensated by the deblocking filter described in this paper. In the post-filter case, the blockiness does not follow the grid boundaries but is spread inside the 4×4 blocks due to blocky reference images that were used for motion compensation [4]. This additionally results in an increased amount of residual coding to remove artificially created high frequency edges and possible ringing effects at low bit rates.

Despite all of these advantages, the requirement of a normative loop filter was extensively debated during the development of the H.264/MPEG-4 AVC standard. A critical factor in the debate was the comparatively high complexity of the loop filter. Even after a tremendous effort in speed optimization of the filtering algorithms, the filter can easily account for one-third of the computational complexity of a decoder. This is true even

Manuscript received May 2, 2003.

P. List is with Deutsche Telekom, T-Systems, 64295 Darmstadt, Germany (e-mail: Peter.List@t-systems.com).

A. Joch is with UB Video Inc., Vancouver, BC V6B 2R9, Canada (e-mail: anthony@ubvideo.com).

J. Lainema and M. Karczewicz are with the Nokia Research Center, Irving, TX 75039 USA (e-mail: jani.lainema@nokia.com; marta.karczewicz@nokia.com).

G. Bjøntegaard is with TANDBERG, N-1324 Lysaker, Norway (e-mail: gbj@tandberg.no).

Digital Object Identifier 10.1109/TCSVT.2003.815175

[1]While this paper describes the deblocking filter design in the final draft of the H.264/MPEG-4 AVC standard [1], results and figures have been generated using the TML 8.5 software, which corresponds to an earlier draft specification [6]. Some parts of the deblocking filter design have been modified in the interim between these drafts (primarily for complexity reduction). However, the major properties and the performance of the filter have remained unchanged.

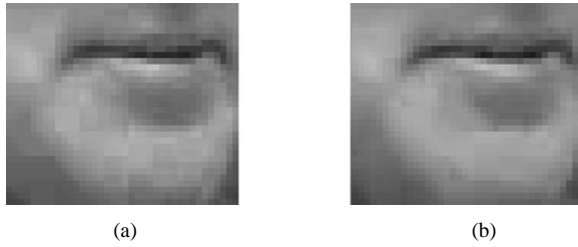(a)   (b)

Fig. 1.   Detail of luminance input to the deblocking filter in the case of (a) loop-filtering and (b) post-filtering based system.

though the loop filter can be implemented without any multiplication or division operations.

The complexity is mainly based on the high adaptivity of the filter, which requires conditional processing on the block edge and sample levels. As a consequence, conditional branches almost inevitably appear in the inner most loops of the algorithm. These are known to be very time consuming and are also quite a challenge for parallel processing in DSP hardware or SIMD code on general-purpose processors.

Another reason for the high complexity is the small block size employed for residual coding in the H.264 coding algorithm. With the 4×4 blocks and a typical filter length of 2 samples in each direction, almost every sample in a picture must be loaded from memory, either to be modified or to determine if neighboring samples will be modified. This was not the case for the H.263 loop filter or any MPEG-4/H.263 post filters, which operate on an 8×8 block structure.

In Sections II–IV, we provide an overview of the design of the H.264/MPEG-4 AVC adaptive deblocking filter. For a complete and detailed description of the filtering process, see [1]. Further information on the evolution of the filter design is available in the many contributions to the H.264/MPEG-4 AVC standardization effort that influenced the final filter design, including [7]–[19].

II. BOUNDARY ANALYSIS

A. Error Distribution in a 4×4 Block

When using a block transform for residual coding, it is well known that the coding errors are larger near the block boundaries than in the middle of the block. The numbers below show an example of the square error distribution over a 4×4 block

$$
\begin{array}{cccc}
122 & 107 & 106 & 111 \\
103 & 102 & 101 & 112 \\
106 & 100 & 98 & 108\,\dot{} \\
118 & 105 & 106 & 120
\end{array}
$$

A heuristic argument for this effect is that an interior sample has several surrounding samples which add "weight" to a good reconstruction—whereas an edge sample has less such weight and thereby obtains a poorer reconstruction. As a result of this uneven error distribution, there is a potential for *objective* quality improvement by block edge filtering. This potential was kept in mind when designing the filter.

The H.264/MPEG-4 AVC deblocking filter is adaptive on several levels.

TABLE I
FILTER STRENGTH PARAMETER AS A
FUNCTION OF CODING MODE

Block modes and conditions	Bs
One of the blocks is Intra *and* the edge is a macroblock edge	4
One of the blocks is Intra	3
One of the blocks has coded residuals	2
Difference of block motion $\geq$ 1 luma sample distance	1
Motion compensation from different reference frames	1
Else	0

- On the slice level, the global filtering strength can be adjusted to the individual characteristics of the video sequence.
- On the block-edge level, filtering strength is made dependent on the inter/intra prediction decision, motion differences, and the presence of coded residuals in the two participating blocks. Special strong filtering is applied for macroblocks with very flat characteristics to remove "tiling artifacts".
- On the sample level, sample values and quantizer-dependent thresholds can turn off filtering for each individual sample.

Sections II-B and II-C describe in detail how this adaptivity is designed into the H.264/MPEG-4 AVC deblocking filter.

B. Edge Level Adaptivity of the Filter

To every edge between two 4×4 luminance sample blocks, a *Boundary-Strength* (Bs) parameter is assigned an integer value from 0 to 4. Table I shows how the value of Bs depends on the modes and coding conditions of the two adjacent blocks. In this table, conditions are evaluated from top to bottom, until one of the conditions holds true, and the corresponding value is assigned to Bs.

In the actual filtering algorithm, Bs determines the strength of the filtering performed on the edge, including a selection between the two primary filtering modes. A value of 4 means a special mode of the filter is applied, which allows for the strongest filtering, whereas a value of 0 means no filtering is applied on this specific edge. In the standard mode of filtering which is applied for edges with Bs from 1 to 3, the value of Bs affects the maximum modification of the sample values that can be caused by filtering. The gradation of Bs reflects that the strongest blocking artifacts are mainly due to intra and prediction error coding and are to a somewhat smaller extent caused by block motion compensation.

The Bs values for filtering of chrominance block edges are not calculated independently, but instead copied from the values calculated for their corresponding luminance edges.

In the case of macroblock adaptive frame/field coding (MBAFF), the conditions in Table I get somewhat more complex because any of the two adjacent blocks might belong to a frame- or a field-coded macroblock. The principle of varying filter strength remains the same in any case. To avoid excessive blurring, special consideration is made to prevent very strong
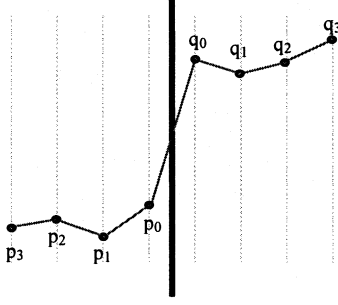
Fig. 2. One-dimensional visualization of a block edge in a typical situation where the filter would be turned on.

filtering of horizontal edges of field-coded macroblocks, since the spatial extent of the vertical filtering is doubled for such macroblocks.

C. Sample-Level Adaptivity of the Filter

In deblocking filtering, it is crucially important to be able to distinguish between true edges in the image and those created by quantization of the DCT coefficients. To preserve image sharpness, the true edges should be left unfiltered as much as possible while filtering artificial edges to reduce their visibility.

In order to separate these two cases, the sample values across every edge to be filtered are analyzed. Let us denote one line of sample values inside two neighboring 4×4 blocks by $p_3, p_2, p_1, p_0, q_0, q_1, q_2, q_3$, with the actual boundary between p_0 and q_0, as shown in Fig. 2. Up to three sample values for luminance and one for chrominance on each side of the edge may be modified by the filtering process.

As stated in Section II-B, filtering does not take place for edges with Bs equal to zero. For edges with nonzero Bs values, a pair of quantization-dependent parameters, referred to as α and β, are used in the content activity check that determines whether each set of samples is filtered. Filtering on a line of samples only takes place if the three conditions

$$|p_0 - q_0| < \alpha(\text{Index}_A) \tag{1}$$
$$|p_1 - p_0| < \beta(\text{Index}_B) \tag{2}$$
$$|q_1 - q_0| < \beta(\text{Index}_B) \tag{3}$$

all hold. In these conditions, both table-derived thresholds α and β are dependant on the average quantization parameter (QP) employed over the edge, as well as encoder selected offset values that can be used to control the properties of the deblocking filter on the slice level. These table index values are calculated as

$$\text{Index}_A = \text{Min}(\text{Max}(0, \text{QP} + \text{Offset}_A), 51) \tag{4}$$
$$\text{Index}_B = \text{Min}(\text{Max}(0, \text{QP} + \text{Offset}_B), 51) \tag{5}$$

where 0–51 represents the range of valid QP values.

The values of α and β are defined approximately according to the following relationships:

$$\alpha(x) = 0.8(2^{x/6} - 1) \tag{6}$$
$$\beta(x) = 0.5x - 7. \tag{7}$$

Thus, in general, $\beta(x)$ is considerably smaller than $\alpha(x)$. To define the actual tables, variations from this basic relationship

have been made based on empirical tests to produce visually pleasing results for a variety of content. In particular, at the low end of the table, values are clipped to zero so that for values of $\text{Index}_A < 16$ or $\text{Index}_B < 16$, one or both of α and β become 0 and filtering is effectively turned off.

The dependency of α and β on QP links the strength of filtering to the general quality of the reconstructed picture prior to filtering. Since the thresholds values increase with QP, boundaries that contain higher content activity are filtered when QP is larger, since the coding error (size of artifacts) increases with QP. The exponential nature of α reflects the dependency on QP of the size of an expected blocking artifact, since the quantization step size doubles every time QP is increased by 6.

D. Slice-Level Adaptivity of the Filter

On the slice level, encoder-selectable offsets—referred to as Offset_A and Offset_B—may be used to adjust the values of α and β used in filtering and thereby increase or decrease the amount of filtering that takes place compared to filtering with the default zero offsets. The offset values are transmitted in the slice header syntax and are applied to the QP-based addressing of the α and β tables.

The ability to control the properties of deblocking filter by transmitting nonzero offsets provides the encoder designer with the ability to optimize the subjective quality of the decoded video beyond that provided by use of the default tables. For example, reducing the amount of filtering by transmitting negative offsets can help to maintain the sharpness of small spatial details, particularly with high-resolution video content, in which small blocking artifacts tend to be less apparent. On the other hand, using positive offsets to increase the amount of filtering can improve subjective quality on content where visible blocking artifacts remain if the default values are used. This is beneficial for lower resolution content with smooth brightness transitions and to remove additional artifacts that might be introduced by sub-optimal motion estimation, mode decisions, or residual coding.

III. FILTERING

A. Overview of Filtering Operations

In order to ensure a perfect match in the filtering process between encoders and decoders, filtering operations must be conducted in a specific order throughout each coded picture. Filtering is conducted "in-place," so that the modified sample values after featuring each line of samples across an edge are used as input values to subsequent operations.

Filtering occurs on a macroblock basis, with horizontal filtering of the vertical edges performed first, followed by vertical filtering (of the horizontal edges). Both directions of filtering on each macroblock must be conducted before moving on to the next macroblock. The macroblocks are filtered in raster-scan order throughout the picture. For MBAFF coded frames, in which pairs of vertically adjacent macroblocks are grouped together, the filtering order is based on these macroblock pairs, with the pairs being filtered in raster-scan order throughout the frame, and the top macroblocks being filtered first within each pair.

For each luminance macroblock, the left-most edge of the macroblock is filtered first, followed from left to right by the three vertical edges that are internal to the macroblock. Similarly, the top edge of the macroblock is filtered first in the horizontal filtering pass, followed by the three internal horizontal edges from top to bottom. Chrominance filtering follows a similar order, with one external edge and one internal edge in each direction for each 8×8 chrominance macroblock.

Two filtering modes are defined and are selected based on the Bs parameter for a set of samples. A special mode of filtering that allows for stronger filtering is applied when Bs is equal to 4; the more common mode of filtering is applied otherwise (Bs = 1, 2, or 3).

For both filtering modes, the β threshold value is used to evaluate two additional spatial activity conditions that are used to determine the extent of the filtering in the case of luminance samples

$$|p_2 - p_0| < \beta(\text{Index}_B) \qquad (8)$$
$$|q_2 - q_0| < \beta(\text{Index}_B). \qquad (9)$$

When these conditions hold true, which occurs in the case of small changes in intensity on either side of the edge, the strength of the filtering is greater.

B. Filtering for Edges With Bs From 1 to 3

For clarity, the filter operations are divided into basic filter operation and clipping.

1) Basic Filter Operation: We first describe the basic filtering operation for luminance. In this mode of filtering, the filtered values p_0' and q_0' are calculated as

$$p_0' = p_0 + \Delta_0 \text{ and} \qquad (10)$$
$$q_0' = q_0 - \Delta_0 \qquad (11)$$

where the value of Δ_0 is calculated in a two-step process, with the calculation of an initial Δ_{0i} value, followed by clipping of this value before it is applied in the above equations.

The initial value, Δ_{0i} is computed based on the sample values across the edge

$$\Delta_{0i} = (4(q_0 - p_0) + (p_1 - q_1) + 4) \gg 3. \qquad (12)$$

The impulse response of these operations for calculating p_0' is $(1, 4, 4, -1)/8$.

The values of p_1 and q_1 are only modified if the corresponding condition (8) or (9) is true. Otherwise, the values are not modified. That is, if condition (8) is true, then the filtered value of p_1' is calculated as

$$p_1' = p_1 + \Delta_{p1}. \qquad (13)$$

Similarly, if condition (9) is true, the filtered value of q_1' is calculated as

$$q_1' = q_1 + \Delta_{q1}. \qquad (14)$$

These values are also calculated in a two-step process. The initial Δ value for computing p_1' is computed as

$$\Delta_{p1i} = (p_2 + ((p_0 + q_0 + 1) \gg 1) - 2p_1) \gg 1. \qquad (15)$$

The value Δ_{q1i} is obtained accordingly, substituting q_2 and q_1 for p_2 and p_1, respectively. The corresponding impulse response (1, 0, 0.5, 0.5)/2 has a very strong low-pass characteristic.

2) Clipping: If these intermediate values Δ_{0i}, Δ_{p1i}, and Δ_{q1i} were used directly in the filtering equations, it would result in far too much low-pass filtering (blurring). A significant part of the adaptivity of the filter is obtained by limiting these Δ values. This process is called clipping. Different procedures for clipping are applied for the interior and edge samples.

The Δ values that are used in filtering the interior samples are clipped the range $-c_1$ to c_1, where c_1 is a parameter that is determined based on a table that is indexed in two dimensions, with the value of Index_A as computed for determining α used in one dimension, and the Bs value used in the other. The value of c_1 increases allowing stronger filtering, as the values of Index_A and Bs increase. The final clipping values for filtering of p_1 and q_1 are calculated as

$$\Delta_{p1} = \text{Min}(\text{Max}(-c_1, \Delta_{p1i}), c_1) \qquad (16)$$
$$\Delta_{q1} = \text{Min}(\text{Max}(-c_1, \Delta_{q1i}), c_1). \qquad (17)$$

For filtering of the edge samples p_0 and q_0, the clipping range that is applied to Δ_{0i} is determined based on the value of c_1 and the evaluation of the conditions (8) and (9). The clipping value c_0 is first set equal to c_1, and then incremented by 1 for each of conditions (8) and (9) that holds true. Then, the amount of modification that will be applied to each of the edge samples is computed as

$$\Delta_0 = \text{Min}(\text{Max}(-c_0, \Delta_{0i}), c_0). \qquad (18)$$

Thus, stronger filtering is applied to the edge samples when the changes in intensity on each side of the edge are smaller than the β threshold (and the p_1 and/or q_1 sample values were also modified).

For chrominance filtering, only the p_0 and q_0 values may be modified. These are filtered in the same way as for luminance, except that the clipping value, c_0 is set equal to c_1 plus 1. In this way, there is no need to evaluate conditions (8) and (9) for chrominance on edges with Bs less than 4, and therefore no need to access sample values p_2 and q_2.

C. Filtering for Edges With Bs Equal to 4

Intra coding in H.264/MPEG-4 AVC tends to use 16×16 luma sample prediction modes when coding nearly uniform image areas. This causes small amplitude blocking artifacts at the macroblock boundaries. However, due to the Mach band effect [20], even very small differences in the intensity values are perceived as abrupt steps in these cases. To compensate for this tiling effect, stronger filtering is applied on boundaries between two macroblocks with smooth image content.

For luminance filtering, a decision is made based on the image content between a very strong 4- and 5-tap filter that modifies the edge sample and two interior samples on each side, or a weaker 3-tap filter modifies only to the edge sample. The stronger filter is only applied when the following constraint on the difference across the edge holds true:

$$|p_0 - q_0| < (a \gg 2) + 2. \qquad (19)$$

TABLE II
AVERAGE BIT RATE SAVINGS FOR ALIGNED PSNRs OBTAINED WITH
TML 8.5 LOOP FILTER

Sequence	Type	Bitrate saving [%]
Container	10 Hz, QCIF	8.9
Foreman	10 Hz, QCIF	4.8
News	10 Hz, QCIF	8.1
Silent Voice	15 Hz, QCIF	9.4
Paris	15 Hz, CIF	7.6
Foreman	15 Hz, CIF	6.9

Fig. 3. Luminance PSNR curve for Foreman with and without a loop filter in TML 8.5.
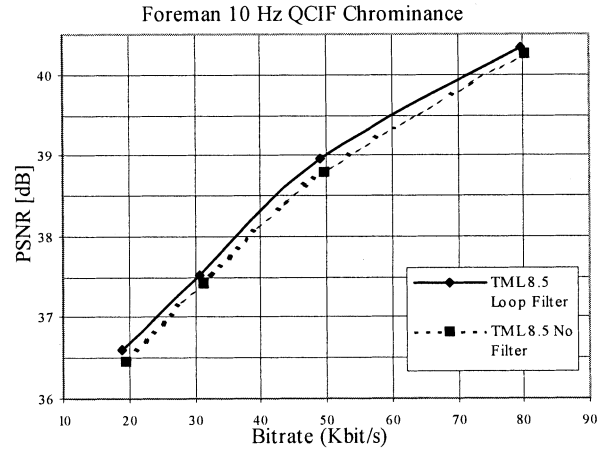

Fig. 4. Chrominance PSNR curve for Foreman with and without a loop filter in TML 8.5.
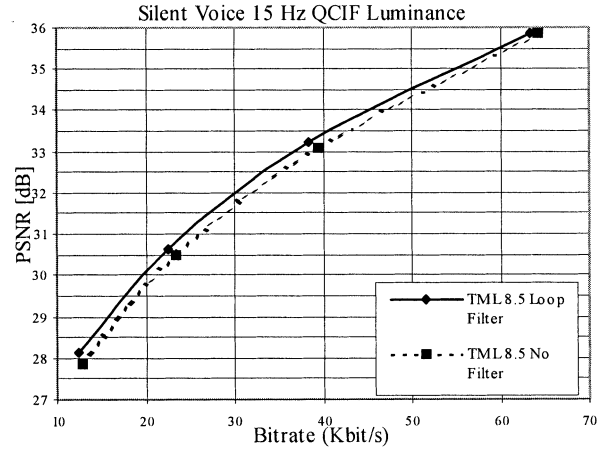

Fig. 5. Luminance PSNR curve for Silent Voice with and without a loop filter in TML 8.5.
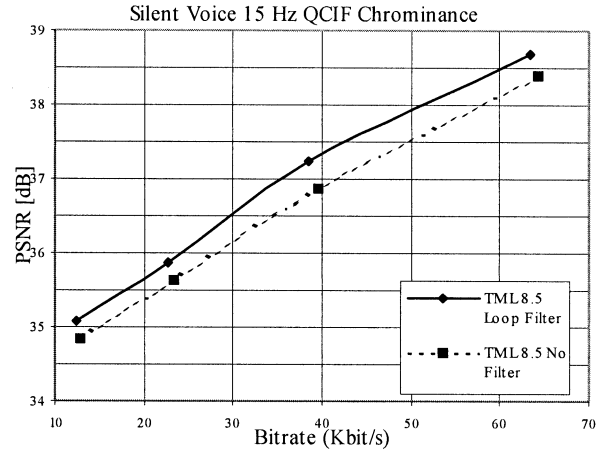

Fig. 6. Chrominance PSNR curve for Silent Voice with and without a loop filter in TML 8.5.

Notice that (19) is a similar condition to (1), with a tighter constraint on the maximum sample value difference across the edge.

For luminance filtering, when the smoothness conditions (8) and (19) hold true, the filtered values are calculated according to the following equations:

$$p'_0 = (p_2 + 2p_1 + 2p_0 + 2q_0 + q_1 + 4) \gg 3 \qquad (20)$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) \gg 2 \qquad (21)$$

$$p'_2 = (2p_3 + 3p_2 + p_1 + p_0 + q_0 + 4) \gg 3. \qquad (22)$$

Otherwise, for chrominance filtering, or if either (8) or (19) is false, only p_0 is modified according to the following equation:

$$p'_0 = (2p_1 + p_0 + q_1 + 2) \gg 2 \qquad (23)$$

and p_1 and p_2 are left unchanged.

The q values are modified in a similar manner, substituting condition (9) for condition (8) when selecting the filter for luminance.

IV. RESULTS

Table II lists examples of bitrate savings on aligned luminance and chrominance PSNRs [21], [22] achieved with the TML 8.5 loop filter against the same H.264/MPEG-4 AVC codec without a deblocking filter.

More specific PSNR curves for Foreman and Silent Voice sequences are given in Figs. 3–6. Even more remarkable are the improvements in subjective picture quality illustrated in Figs. 7

and 8, which compare loop-filter output images to unfiltered output images at two different bit rates and resolutions.

V. CONCLUSIONS

The adaptive deblocking filter described in this paper achieves substantial objective and subjective quality im-

Fig. 7. Detail of the luminance output in the case of (a) loop filtering and (b) no filtering. CIF sequence was coded at 200 kbps and 15 fps.

Fig. 8. Detail of the luminance output in the case of (a) loop filtering and (b) no filtering. QCIF sequence was coded at 30 kbps and 10 fps.

provements with a reasonably simple algorithm. The good performance is based on reliable detection of real and artificially created edges and efficient filtering of the latter ones. Bit-rate savings exceeding 9% are observed with equal PSNR levels together with significantly improved visual quality.

REFERENCES

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC), Mar. 2003.
[2] K. K. Pang and T. K. Tan, "Optimum loop filter in hybrid coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 158–167, Apr. 1994.
[3] Y.-L. Lee and H. W. Park, "Loop filtering and post-filtering for low-bit-rates moving picture coding," *Signal Processing: Image Commun.*, vol. 16, pp. 871–890, 2001.
[4] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 156–160, Feb. 1999.
[5] J. Lainema and M. Karczewicz, "TML 8.4 Loop Filter Analysis," ITU-T SG16 Doc. VCEG-N29, 2001.
[6] (2001) H.26L Test Model Long Term Number 8 (TML-8). [Online]. Available: ftp://standard.pictel.com/video-site/h26L/
[7] J. Lainema and M. Karczewicz, "Core Experiment Results on Low Complexity Loop Filtering," ITU-T SG16 Doc. VCEG-M21, 2001.
[8] ——, "Further Improvements on TML Loop Filtering," ITU-T SG16 Doc. VCEG-M22, 2001.
[9] G. Bjøntegaard and I. Lille-Langoy, "Possible Simplifications of the Present Deblocking Filter in TML 5.9," ITU-T SG16 Doc. VCEG-M30, 2001.
[10] P. List, "Proposal for a Simplification of the H.26L Loopfilter," ITU-T SG16 Doc. VCEG-M48, 2001.
[11] ——, "Report of the Ad Hoc Committee on Loop Filter Improvement," ITU-T SG16 Doc. VCEG-N08r1, 2001.
[12] S. Sun and S. Lei, "Improved TML Loop Filter With Lower Complexity," ITU-T SG16 Doc. VCEG-N17, 2001.
[13] G. Côté, L. Winger, and M. Gallant, "Lower Complexity Deblocking Filter With In-Place Filtering," Doc. VCEG-O39, 2001.
[14] P. List, "AHG Report: Loop Filter," Doc. JVT-B011r2, 2002.
[15] J. Au, B. Lin, A. Joch, and F. Kossentini, "Complexity Reduction and Analysis for Deblocking Filter," Doc. JVT-C094, 2002.
[16] A. Joch, "Improved Loop-Filter Tables and Variable-Shift Table Indexing," Doc. JVT-D038, 2002.
[17] ——, "Loop Filter Simplification and Improvement," Doc. JVT-D037, 2002.
[18] C. Gomila and A. Joch, "Simplified Chroma Deblocking (Revisited)," Doc. JVT-E089, 2002.
[19] A. MacInnis and S. Zhong, "Corrections to Loop Filter in the Case of MB-AFF," Doc. JVT-F027, 2002.
[20] A. K. Jain, *Fundamentals of Digital Image Processing*. New York: Prentice-Hall, 1989.
[21] G. Bjøntegaard, "Calculation of Average PSNR Differences Between RD-Curves," ITU-T SG16 Doc. VCEG-M33, 2001.
[22] ——, "Recommended Simulation Conditions for H.26L," ITU-T SG16 Doc. VCEG-M75, 2001.

Peter List was born in 1957. He graduated in physics in 1985 and received the Ph.D. degree in applied physics in 1989, both from the University of Frankfurt/Main, Germany.

He is project manager at T-Systems Nova, Darmstadt, Germany, a reserach and development company of Deutsche Telekom. Since 1990, he has been with Deutsche Telekom, and has actively followed the international standardization of video compression in ISO, ITU, and several European projects for more than ten years.

Anthony Joch received the B.Eng. degree in computer engineering from McMaster University, Hamilton, ON, Canada, in 1999, and the M.A.Sc. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2002.

In 2000, he joined UB Video Inc., Vancouver, BC, where he is currently a Senior Engineer involved in the development of software codecs for the H.264/MPEG-4 AVC standard. His research interests include reduced-complexity algorithms for video encoding, video pre- and post-processing, and multimedia systems. He has been an active contributor to the H.264/MPEG-4 AVC standardization effort, particularly in the area of deblocking filtering and as a co-chair of the ad-hoc group for bitstream exchange.

Jani Lainema received the M.Sc. degree in computer engineering from Tampere University of Technology, Tampere, Finland, in 1996.

He joined the Visual Communications Laboratory of Nokia Research Center, Irving, TX, in 1996, where he is currently a Project Manager and Senior Research Scientist. His research interests include video, image and graphics coding and communications.

Gisle Bjøntegaard received the Dr. Phil.degree in physics from the University of Oslo, Oslo, Norway, in 1974.

From 1974 to 1996, he was a Senior Scientist with Telenor Research and Development, Oslo, Norway. His areas of research included radio link network design, reflector antenna design and construction, digital signal procession, and development of video compression methods. From 1996 to 2002, he was a Group Manager at Telenor Broadband Services, Oslo, Norway, where his areas of work included the design of point-to-point satellite communication and development of satellite digital TV platform. Since 2002, he has been a Principal Scientist at Tandberg Telecom, Lysaker, Norway, working with video-coding development and implementation. He has contributed actively to the development of the ITU video standards H.261, H.262, H.263, and H.264, as well as to ISO/IEC MPEG2 and MPEG4.

Marta Karczewicz received the M.S. degree in electrical engineering in 1994 and Dr. Technol. degree in 1997, both from Tampere University of Technology, Tampere, Finland.

During 1994–1996, she was Researcher in the Signal Processing Laboratory, Tampere University of Technology. Since 1996, she has been with the Visual Communication Laboratory, Nokia Research Center, Iriving, TX, where she is currently a Senior Research Manager. Her research interests include image compression, communication and computer graphics.