Weighted Temporal Long Trajectory Filtering for Video Compression

Marko Esche, Alexander Glantz, Andreas Krutz, Michael Tok, and Thomas Sikora
Technische Universität Berlin
Communication Systems Group, Sekr. E-N 1
Einsteinufer 17, D-10587 Berlin, Germany
Email: {esche,glantz,krutz,tok,sikora}@nue.tu-berlin.de

Abstract—In the context of the HEVC standardization activity, in-loop filters such as the adaptive loop filter and the deblocking filter are currently under investigation. Both filters work in the spatial domain only, despite the temporal correlation within video sequences. In this work we previously introduced filter, that uses temporal information for deblocking and denoising instead, is integrated into the HEVC test model HM 3.0. It is shown how the filter is to be adapted to work in combination with the adaptive loop filter for the HEVC low-delay profile. In addition, an optimal weighting function for the filtered luma samples based on the quantization parameter is derived. Bit rate reductions of up to 7.6% are reported for individual sequences.

I. INTRODUCTION

Ever since the standardization of H.264/AVC it has been a well-known fact that in-loop filters can be used to improve both the objective and subjective quality of a decoded video sequence through noise reduction. In [1], List et al. identified the main two sources of the noise introduced during the encoding process, namely block-based motion estimation (ME) and the quantization of DCT coefficients. In the current test model of the HEVC standardization activity HM [2] two in-loop filters are employed. The first is the deblocking filter (DF) originally described in [1], the second is the Wiener-based adaptive loop filter ALF [3]. The ALF can be easily adapted to changing video content through the use of varying filter kernels and the control of the filtered image regions by use of a quadtree partitioning algorithm. Despite the fact that ME adds a temporal component to the noise in the encoded sequence, both filters use information from one frame only and thus remain inherently spatial filters. Attempts to overcome this deficiency were, for instance, described in [4] and [5]. In [4] temporal filtering along motion trajectories was performed as an analysis/synthesis step during 3-D subband coding, which combines both prediction and noise reduction. The global motion temporal filter detailed in [5] performs background filtering by warping neighboring frames into the current one through the use of homographies transmitted in the bitstream. In [6], the authors described the temporal trajectory filter (TTF) that makes use of individual pixel trajectories derived from the sequence’s motion vectors to perform temporal filtering. In this context, a pixel trajectory is defined as the locations through which a certain image point moves over the duration of a video sequence. For a given dataset, bit rate reductions of up to 12% were reported for the H.264/AVC baseline profile. In this paper, the TTF is investigated in the environment of the HM. In addition, it is shown how the length of individual pixel trajectories can be increased to achieve better filtering performance. The remainder of the paper is structured as follows. Section II derives a mathematical model for the noise encountered along a pixel trajectory, if the trajectory is completely known. Based on this model, optimal filter coefficients are calculated. In Section III, three previously described thresholds are revisited that allow the calculation of pixel trajectories directly from the motion vectors conveyed in the bit stream. In addition, it is shown how the filter needs to be modified to extend the length of the temporal trajectories. Section IV presents the experimental evaluation conducted in the context of the HM. To this end, the filter’s performance is also compared against the ALF and a combination of both filters is presented. Section V summarizes the paper.

II. THEORETICAL BASIS

For any given pixel \((x_0, y_0)^T\) in frame \(j\) it is assumed that its locations \((x_i, y_i)^T, 1 \leq i < N, in N – 1\) previous frames are also known. If \(Y_n(x, y)\) denotes the luminance component of frame \(n\) at location \((x, y)^T\) the distorted versions of the original sample \(Y_j(x_i, y_i)\) in any of the \(N – 1\) previous frames given by

\[
Y_{j-i}(x_i, y_i) = Y_j(x_0, y_0) + n_i, 1 \leq i < N. \tag{1}
\]

Even though the motion of the pixel is perfectly known, a noise term \(n_i\) with variance \(\sigma_i\) is introduced due to the reduced quality of the encoded sequence. As described in [6], it can be assumed that all \(n_i\) are uncorrelated. A filtered version of the original luma component can then be computed by calculating a weighted mean

\[
Y^*_j(x_0, y_0) = \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot Y_{j-i}(x_i, y_i)
= \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot Y_j(x_0, y_0) + \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot n_i \tag{2}
= \frac{1}{N} Y_j(x_0, y_0) \sum_{i=0}^{N-1} \beta_i + \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot n_i.
\]

Where \(\beta_i\) are the individual weights per frame with \(\sum_{i=0}^{N-1} \beta_i = N\) to make the filter unbiased. This leads to the
The variance of the filtered noise \( \hat{n}_j \) is subsequently given by

\[
\sigma_{\hat{n}}^2 = E \left[ \hat{n}_j \hat{n}_j \right] = \frac{1}{N^2} \sum_{i=0}^{N-1} \beta_i \sigma_n^2.
\]

As the filter is to minimize \( \sigma_{\hat{n}}^2 \) constraint to \( \sum_{i=0}^{N-1} \beta_i = N \), the minimum may be found by Lagrangian minimization

\[
\frac{\partial}{\partial \beta_i} \left( \sum_{i=0}^{N-1} \beta_i \sigma_n^2 - \lambda \left( N - \sum_{k=0}^{N-1} \beta_k \right) \right) = 0
\]

\[
2\beta_i \sigma_n^2 + \lambda = 0 \Rightarrow \beta_i = -\frac{\lambda}{2\sigma_n^2}.
\]

\[
\sum_{k=0}^{N-1} \beta_k = N \Leftrightarrow \sum_{k=0}^{N-1} \frac{\lambda}{2\sigma_n^2} = N
\]

\[
\beta_i = -\frac{\lambda}{2\sigma_n^2} \Rightarrow \frac{\lambda}{\sum_{k=0}^{N-1} \frac{1}{\sigma_n^2}} = \frac{N}{\sigma_n^2}.
\]

In theory, the reconstruction error variance \( \sigma_{\hat{n}}^2 \) for every pixel along the trajectory would be required to calculate the optimal filter weight \( \beta_i \). According to Wiegand and Girod [7] the distortion variance in a reconstructed frame is given by

\[
D_{REC} = \frac{Q_{step}^2}{3}
\]

with zero mean.

Where \( Q_{step} \) is the quantizer step size selected by the quantization parameter QP. Both in H.264/AVC and in HEVC, \( Q_{step} \) is roughly

\[
Q_{step} = 0.625 \cdot 2^{\frac{QP}{6}}.
\]

Subsequently, the optimal filter weight for frame \( i \) according to its QP may be calculated

\[
\sigma_i^2 = D_{REC,i} = \frac{1}{3} \left( 0.625 \cdot 2^{\frac{QP_i}{6}} \right)^2
\]

\[
\beta_i(QP_i) = 3 \cdot \left( 0.625 \cdot 2^{\frac{QP_i}{6}} \right)^{-2}.
\]

In the case of the H.264/AVC baseline profile this yields identical weights for every frame. When varying QPs are used as in the HEVC low-delay setting, the optimal weights can be calculated at the decoder requiring no additional side information.

III. FILTER DESIGN

In the low-delay high efficiency setting of HEVC with an IBBB coding structure every B-predicted block can have up to two motion vectors pointing to one of the last four encoded pictures. It is assumed that the motion vector for a given block also describes the individual motion of every pixel within the block. The components of the two resulting motion vector fields for frame \( i \) shall be denoted by \((dx_{i,0},dy_{i,0})^T\) and \((dx_{i,1},dy_{i,1})^T\). Starting again with pixel \((x_0,y_0)^T\) in frame \( i \), two possible locations of the pixel in the referenced frames are therefore given by

\[
\begin{align*}
(x_1, y_1) &= (x_0, y_0) + (dx_{i,0}, dy_{i,0})(x_0, y_0) \\
(x_2, y_2) &= (x_0, y_0) + (dx_{i,1}, dy_{i,1})(x_0, y_0)
\end{align*}
\]

Figure 1 shows how the concatenation of motion vectors is used to derive possible pixel locations over a GOP of four frames. However, not all of these describe the true motion of the pixel. It becomes therefore necessary, to discard those motion vectors that have purely been chosen due to rate-distortion optimization and thus may not relate to the true motion of pixels. To this end three thresholds are used. In each of the following equations the motion vectors are scaled according to the temporal distance that they span.

A. Absolute Error Along the Trajectory

For every pixel the absolute difference of two consecutive luminance samples \( \Delta Y_i = Y_{i+1} - Y_i \) together with the respective chrominance differences \( \Delta U_i \) and \( \Delta V_i \) are calculated. A sudden change in one of these differences is assumed to indicate that a motion vector no longer describes the true motion of a pixel. The trajectory is only continued, if

\[
\Delta Y_i < T, \Delta U_i < T, \Delta V_i < T \quad T = \begin{cases} 
2T_Y, QP < 30 \\
4T_Y, QP \geq 30
\end{cases}
\]

for a given threshold \( T_Y \), \( 0 \leq T_Y \leq 7 \).

B. Temporal Motion Consistency

In addition, the similarity of consecutive motion vectors is tested. A trajectory is expected to be correct as long as its motion does not change significantly over time. When examining a new motion vector for list \( 0 \) for any given pixel of the trajectory, its Euclidean distance to the vector pointing to
thresholds are simply omitted. A tenth bit can be used to disable the filter
the minimum mean square error is then selected. Each of the
neously at the encoder. The parameter combination yielding
E. Parameter Calculation
calculation of
the respective chrominance samples are now used for the
question from the filtering process and continues the trajectory.
Both the scaling of motion vectors and the block-vote metric
empirically and proved to be well suited for all sequences.

D. Long Trajectories

In the previous implementation described in [6], a trajectory
was interrupted as soon as the luminance difference became
bigger than the threshold \( T_Y \). This makes the coder biased
towards shorter trajectories. In theory, however, the quality of
the filtering process is increased with the number of samples
used. The new design, therefore, does not stop the trajectory
formation all together, but simply omits the luma sample in
question from the filtering process and continues the trajectory.
For the threshold \( T_Y \) the last filtered luminance samples and
the respective chrominance samples are now used for the
calculation of \( \Delta Y_i, \Delta U_i, \) and \( \Delta V_i \).
E. Parameter Calculation

All possible parameter combinations can be tested simulta-
neously at the encoder. The parameter combination yielding
the minimum mean square error is then selected. Each of the
thresholds is transmitted to the decoder requiring 9 additional
bits per frame. A tenth bit can be used to disable the filter
for the current frame all together, in which case the other
thresholds are simply omitted.

Fig. 2. On the left the original motion vectors for all \( 4 \times 4 \) blocks surrounding
the current location are shown. Vectors that span a temporal
distance greater than 1 are marked in gray. After the scaling (right) the \( BV \)-
metric (i.e. the number of vectors differing from the current one) for the
current block is 5.

\[
\sqrt{(dx_{i0} - dx_i)^2 + (dy_{i0} - dy_i)^2} \leq T_{TC}
\]

(12)

with \( 0 \leq T_{TC} \leq 7 \) in quarter-pel, where \( dx_{i0} \) and \( dy_{i0} \) are
the components of the motion vector for list 0 pointing from
the current frame to reference frame \( r \). The temporal motion
consistency is also checked for the vectors of reference list 1.

C. Spatial Motion Consistency

Aside from a temporal comparison of motion vectors,
spatial similarity is also examined, which is a measure for
the reliability of a motion vector. At every frame the motion
vector for a pixel of the trajectory is compared with its eight
neighbors on \( 4 \times 4 \) block level. In this context the block-vote
metric \( BV \) denotes the number of neighboring motion vectors
that differ significantly from the current one. The allowed
maximum difference is \( 30\% \) of the original motion vector’s
length or at least 0.3 quarter-pel. Both values were chosen
empirically and proved to be well suited for all sequences.

Both the scaling of motion vectors and the block-vote metric
are illustrated by Figure 2. The filtering along the trajectory is
continued only, if the block-vote metric for the current pixel
satisfies

\[
BV_{i,0}(x, y) \leq T_{SC}, \quad 0 < T_{SC} \leq 8.
\]

(13)

The TTF has been integrated into the HEVC test model
HM 3.0. Up to 32 previously decoded unfiltered frames are
kept in a buffer to be used for the trajectory formation. The
resulting encoder is depicted in Figure 3, where the TTF is
used after the deblocking filter. In this setting the ALF (dotted
connections) was disabled. Tests have been conducted for a
variety of sequences listed in Table I. The exact configuration
for the low-delay high efficiency setting may be found in [2].

The HM 3.0 without the ALF is compared against the HM
with the added TTF using the Bjøntegaard metric described
in [8]. For comparison, the individual gain provided by the
ALF per sequence was also calculated. The resulting BD-rates
for both filters may be found in Table I in columns 4 and
5. For all tested sequences except Vidyo3 the TTF produces
a bit rate reduction. For Vidyo3 the increase of 0.07% is,
however, only very slight. Nevertheless, the ALF produces a
higher gain than the TTF for all tested sequences. This is only
to be expected, as the ALF has undergone many significant
improvements over the last years. Even though, both filters
produce similar gains both for \( BQSquare \) and for \( Waterfall \).
The average bit rate reduction produced by the TTF for the
dataset is 1.5%. The true potential of both filters can be
exploited when both are used in combination. In this setting,
the average encoding time is increased by 190% compared to
the HEVC encoder with the ALF. The decoder complexity,
however, is only increased by about 30%. In order to show
that ALF and TTF are not mutually exclusive, columns 8 and
9 of Table I compare HM 3.0 with and without ALF against
the combination of ALF and TTF. In this case, a modified
encoder as depicted in Figure 3 with the dotted connections
is used. The combination of both filters outperforms the test
model HM 3.0 for almost all sequences. For \( BQSquare \) and
\( BlowingBubbles \) in particular, the gain produced by both filters
together equals the sum of their individual quality improve-
m ents. A possible explanation for this finding may be different
noise sources that are compensated separately by both filters,
so that there is no unintended interference between the two
approaches. In this combination, the TTF provides an average
BD-rate of \(-1.4\% \) when compared with the HEVC low-delay

Fig. 3. The TTF is included in the local decoder loop of the encoder after
the deblocking filter. For the first test the ALF was disabled. When both ALF
and TTF are used together, the respective frame in the TTF’s buffer is updated
after the ALF has been applied.

IV. EXPERIMENTAL EVALUATION

The TTF has been integrated into the HEVC test model
HM 3.0. Up to 32 previously decoded unfiltered frames are
kept in a buffer to be used for the trajectory formation. The
resulting encoder is depicted in Figure 3, where the TTF is
used after the deblocking filter. In this setting the ALF (dotted
connections) was disabled. Tests have been conducted for a
variety of sequences listed in Table I. The exact configuration
for the low-delay high efficiency setting may be found in [2].

The HM 3.0 without the ALF is compared against the HM
with the added TTF using the Bjøntegaard metric described
in [8]. For comparison, the individual gain provided by the
ALF per sequence was also calculated. The resulting BD-rates
for both filters may be found in Table I in columns 4 and
5. For all tested sequences except Vidyo3 the TTF produces
a bit rate reduction. For Vidyo3 the increase of 0.07% is,
however, only very slight. Nevertheless, the ALF produces a
higher gain than the TTF for all tested sequences. This is only
to be expected, as the ALF has undergone many significant
improvements over the last years. Even though, both filters
produce similar gains both for \( BQSquare \) and for \( Waterfall \).
The average bit rate reduction produced by the TTF for the
dataset is 1.5%. The true potential of both filters can be
exploited when both are used in combination. In this setting,
the average encoding time is increased by 190% compared to
the HEVC encoder with the ALF. The decoder complexity,
however, is only increased by about 30%. In order to show
that ALF and TTF are not mutually exclusive, columns 8 and
9 of Table I compare HM 3.0 with and without ALF against
the combination of ALF and TTF. In this case, a modified
encoder as depicted in Figure 3 with the dotted connections
is used. The combination of both filters outperforms the test
model HM 3.0 for almost all sequences. For \( BQSquare \) and
\( BlowingBubbles \) in particular, the gain produced by both filters
together equals the sum of their individual quality improve-
m ents. A possible explanation for this finding may be different
noise sources that are compensated separately by both filters,
so that there is no unintended interference between the two
approaches. In this combination, the TTF provides an average
BD-rate of \(-1.4\% \) when compared with the HEVC low-delay
profile with ALF enabled. For comparison column 12 of Table I also shows the BD-rate produced by TFF and ALF if no long trajectories and a simple average instead of a weighted mean are used. The average BD-rate for the simplified filter is only \(-0.9\%\), which provides evidence for the effectiveness of the weighted filtering. An exemplary RD-curve for the BQS\(\text{quare}\) sequence is given in Figure 4. Below 500kbit/s the TFF performs better than the ALF. For all depicted QPs the combination of both filters outperforms the simple ALF. Apart from objective quality improvements the TFF also increases the visual quality of the decoded video. A part of an exemplary decoded frame from the Waterfall sequence is shown in Figure 5. All decoded sequences together with the respective RD-curves may be found on the accompanying website www.nue.tu-berlin.de/research/wtltf.

V. SUMMARY

The main objective of this work was to demonstrate the possibility of further improving the HEVC test model through the use of a temporal filtering approach. In combination with the optimal sample weighting described in Section II, the proposed

![Fig. 4. RD-curves for all three tested settings for the BQS\(\text{quare}\) sequence, QP 22 to 37.](image)

![Fig. 5. Exemplary decoded frames from the Waterfall sequence.](image)