ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

3D-IN LOOP FILTER FOR VIDEO COMPRESSION BASED ON MOTION COMPENSATED TEMPORAL PIXEL TRAJECTORIES

E. A. Ayele*, S. B. Dhok

* Department of Electronics & Communication Engineering (Center for VLSI & Nanotechnology), Visvesvaraya National Institute of Technology, Nagpur-440010, India

Department of Electronics & Communication Engineering (Center for VLSI & Nanotechnology), Visvesvaraya National Institute of Technology, Nagpur-440010, India

DOI: 10.5281/zenodo.59117

KEYWORDS: Deblocking filter, In-loop filtering, Motion-compensated filtering, Pixel trajectories, Video compression.

ABSTRACT

For H.264 standardization, the use of a deblocking filter was introduced in order to improve the quality of video compression through blocking artifacts reduction. Subsequently, the in-loop filters such as deblocking filter, sample adaptive offset, and adaptive loop filter are launched for the present HEVC standardization. However, both filters use spatial data from a frame of the video sequence only despite the fact of temporal correlation within frames. Hence, the quality of filtering process is restricted. Therefore to conquer this problem, we can apply 3D-filter; the three dimensions being the two spatial coordinate and the time. The intention of this paper is to explore the effectiveness of a new in-loop filter that combines both spatial and temporal information, through quality improvement. The filter only requires a small overhead while using the temporal information transmitted in the bit stream to rebuild the individual motion trajectory of every pixel in a frame at the codec. This temporal information is then accustomed to performing pixel-wise motion compensated temporal filtering. The paper describes a nonlinear temporal filtering algorithm using motion estimation and compensation for reducing noise in video sequences. The background of the filtering technique concept is described and simulation results on several video sequences are presented. The new filter has been integrated with deblocking filter and it is revealed that the filter achieves better performance than the state-of-the-art codec H.264/AVC, which uses a deblocking filter as in-loop filtering over a large range of sequences and bit rates.

INTRODUCTION

Noise reduction in video sequences is possible to the extent that image and noise components have different characteristics. A major unique feature between the noise and signal in video sequences is that the noise is uncorrelated from frame to frame, while the image is highly correlated, especially in the direction of motion. By performing a low-pass temporal filtering in the direction of motion, the noise component can be attenuated without affecting the signal component. For stationary random processes, the classical method of noise reduction is Wiener filtering, based on the image and noise power spectra [1]. However, images are not well modelled by stationary random processes, and other approaches based on improved image models are required.

In block-based motion compensated video codec, block-artifacts are introduced at low bit rates due to the combination of motion compensated (MC) prediction and quantization of DCT coefficients [2]. The main difference between both noise sources is that MC is performed in the temporal domain on a number of reference frames, while the quantization of the DCT coefficients concerns the spatial domain of a single video frame only. During the standardization of H.264/AVC, it has been a well-established fact that in-loop filters can improve the subjective quality of the decoded video sequence while at the same time reducing the bit rate needed to transmit the sequence at a predefined quality.

In the context of the next-generation video codec; high-efficiency video coding (HEVC) standardization activity [3], three in-loop filters such as the deblocking filter (DF) [2], the Sample Adaptive Offset algorithm (SAO) [4], and the Wiener-based Adaptive Loop Filter (ALF) [5] have recently received significant attention. The first, deblocking filter (DF) which performs 1-D filtering operations at block boundaries of a single decoded picture

based on the boundary strength and the use of prediction modes. Even though the information from neighboring blocks and their reference frames is also used, the deblocking filter remains an inherently spatial filter. The filter performs well when used as an in-loop filter within the codec and produces average bit rate reductions of 7.0% for a given data set [2]. The second is the Sample Adaptive Offset algorithm (SAO) which improves the quality of reconstructed frames by sorting each decoded pixel into a number of categories. In accordance with the selected category, a simple counterbalance is then added to the pixel. A bit rate reduction of 2% produced by the SAO is reported for the low delay high efficiency setting of HEVC. The third filter is The Wiener based filtering approach incorporated in HEVC, which is called an adaptive loop filter (ALF) proves to be more adaptive to an individual video sequence, as its parameters are optimized to provide a maximum coding gain on a frame-by-frame basis [5]. The ALF applies a two-dimensional filter kernel of 5 x 5, 7x 7 or 9 x 9 pixels to certain image regions. As reported in [6], the filter produces an average bit rate reduction of 6.21% for the common coding conditions defined in [7]. Although block artifacts are produced by transferring blocks of pixels from reference into current frame, most in-loop filters only make use of spatial information from the current frame itself. None of the filters mentioned so far, however, perform filtering in the temporal domain and do not necessarily reduce the block flickering present at low bit rates.

To compensate this deficiency, a novel in-loop filter based on temporal pixel trajectories is proposed [8]. Where, the pixel trajectory is defined as 2D-locations through which a certain image point moves from frame to frame. These systems use motion detection by assessment of the displacement field. The problem of estimating the displacement field is called the correspondence problem. Usually, there are two most important classes of techniques for solving the correspondence problem [9]: matching techniques and spatio-temporal gradient techniques. The matching approach is performed using search techniques in the previous frame, which best matches a given area in the current frame. The measure for best match is usually maximum correlation or minimum absolute difference. The spatio-temporal gradient techniques are derived from the constraint equation relating the spatial gradient to the temporal derivative for moving object [10].

The concept of motion-compensated temporal trajectory filtering has been described by Huang and Hsu [11]. The notation of a motion trajectory and its application to video coding were first described in [12], which is extended to a temporal filtering approach in [13]. Approaches with a similar aim have been presented in [14] and [15], although neither fully utilizes the possibility of identify each pixel with a motion of its own. In all approaches, the displacement at each picture element is estimated, and a temporal averaging is performed along the trajectory of motion. Reference [11] depicts linear non-recursive and median temporal filters, both with and without motion compensation. However, the potential of noise lessening which can be achieved with low-order non-recursive filters is quite limited. In this paper, the nonlinear recursive filtering approach of [16] is extended by the application of motion compensation techniques. Simulation results of its performance on several video sequences are presented. It is shown that this approach is successful in improving video quality, while also improving the performance of subsequent video coding operations. It is also revealed that deblocking filter (DF) performs well when integrated with the temporal trajectory filter (TTF), providing significant bit rate savings for a wide range of sequences.

In this paper, the theory of motion-compensated temporal filtering is discussed in more detail in section 2. Section 3 describes the details on the formation of temporal pixel trajectories and noise reduction through temporal filtering. The subsequent codec design is described in Section 4 together with an algorithm to efficiently compute optimal filter parameters. Section 5 presents the experimental evaluation of the filter and compares its performance with the H.264/AVC deblocking filter. Finally, Section 6 summarizes this paper and provides conclusions.

THEORETICAL FOUNDATIONS OF MOTION-COMPENSATED TEMPORAL FILTERING

Let Y(x, t) be the image intensity at spatial location x = (x_1, x_2) and time t, and let d(x, *t)* be the displacement of the image point at (x, t) between time t - *T* and t. The vector field d(x, *t)* is called the *displacement field* [1]. If the intensity of the object point has not changed over the time *T*, then

$$Y(x,t) = Y(x - d(x,t), t - T) \qquad (1)$$

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

Note that ***d*** is not defined in newly exposed regions, that is, for those pixels which were not visible in the previous field. For background and stationary objects, d(x, t) =0, while for an object in translational motion, d(x, t) is a constant over the object. In general, d(x, t) is a slowly varying function of space, except for discontinuities at the edges of moving objects.

The value over time of the video sequence at a given *object point* forms a one-dimensional signal, defined on the time interval for which this point is visible in the scene. This signal is assumed to be the sum of an image component and an additive noise component. The variation in the image component is solely due to change in the luminance of the object point, caused by changes in illumination or orientation of the object. This change is relatively slow, so that the image component is a low bandwidth signal. The noise is assumed to be white and uncorrelated with the signal. By performing a low pass filtering operation on this signal, the noise component can be significantly attenuated, with a minimal effect on the image component [1].

In practice, the image sequence is sampled spatially, and it is not precisely possible to filter the sequences corresponding to given object points. However, the principle of performing a temporal filtering or averaging operation along the trajectory of motion is feasible. This filtering can be of either the recursive or non-recursive type. Since greater selectivity can be obtained for a given filter order with recursive filters, this type of filter has been chosen for this application. This is especially important in temporal filtering, where each increase by one in filter order requires an additional frame memory.

A block diagram of a first-order recursive temporal filter with motion compensation is shown in Fig. 1 (assume for now that the output of the block nonlinear is a constant value α).

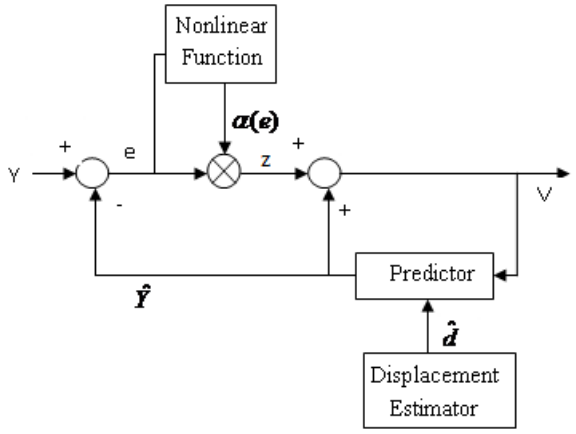


Fig. 1. First-order recursive temporal filter with motion compensation.

The basic operation of this filter is described by:

$$V(x,t) = \alpha Y(x,t) + (1-\alpha)\tilde{V}(x-\hat{d}(x,t),t-T) \qquad (2)$$

where V is the output of the filter, $\hat{d}$ is an estimate of d, and $\tilde{V}$ is an estimate of the output obtained by spatial interpolation. The signal $\hat{Y}(x,t) = \tilde{V}(x-\hat{d},t-T)$ is called prediction and $e = Y - \hat{Y}$ is called the prediction error.

This filter requires a frame memory in order to be able to form the prediction. A module for estimating the displacement field is also required. The displacement field (or optical flow) is the displacement between successive frames of each pixel in an image sequence forms a 2-D vector field [9]. It is estimated based on a constraint equation, which describes the spatial gradient to a temporal directional derivative for a moving object. The estimate for a given pixel is obtained by updating the estimate for the corresponding picture element in the previous frame, using information in a 3-D block centered on the picture element of interest. This estimation can also be performed using any of a number of motion estimation algorithms. The displacement estimator can use the input signal to perform the estimate.

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

An indication of the ability of this filter to reduce noise can be obtained by considering its performance in stationary areas where $d = 0$. In this case, the filter reduces to a standard one-dimensional temporal recursive filter with transfer function [1]:

$$H(z) = \frac{\alpha}{1-(1-\alpha)z^{-1}} \qquad (3)$$

Due to the spatial interpolation error, the performance in moving areas will be slightly different, even if the displacement estimate is perfectly accurate. This can be accomplished by varying the value of α as a function of the prediction error, which is equivalent to passing the prediction error e through a memoryless non-linearity $z = \alpha(e).e$. A typical piecewise-linear characteristic for the function α(e) is given by:

$$\alpha(e) = \begin{cases} \alpha_b, & \text{if } |e| \le P_b; \\ \frac{\alpha_b - \alpha_e}{P_b - P_e}|e| + P_b\alpha_e - P_e\alpha_b, & \text{if } P_b < |e| \le P_e; \\ \alpha_e, & \text{if } |e| > P_e \end{cases} \qquad (4)$$

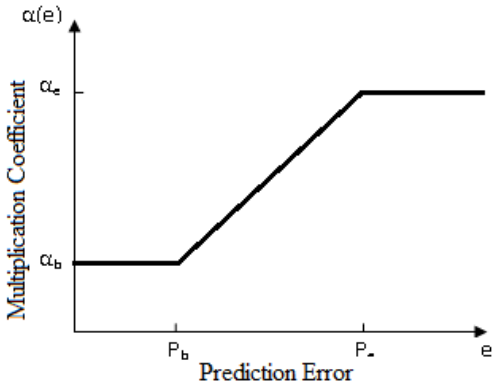


Fig. 2. Nonlinear function for multiplier coefficient α.

As it is shown in Fig.2, in areas where the motion is tracked and e(x,t) is small (of the order of the noise level), a linear temporal filtering with Parameter $\alpha=\alpha_b$ is performed. In areas where the motion is not being tracked and e(x, t) is large, a temporal filtering with Parameter α_e is performed. To avoid introducing artifacts in these regions, α_e is typically set to unity. For values of e between P_b and P_e, α(e) varies linearly between α_b and α_e to provide a smooth transition between regions where motion is tracked and where it is not. The choice of values of P_b and P_e to be used depends on the noise level and the appearance of artifacts.

CONSTRUCTION OF MOTION-COMPENSATED TEMPORAL PIXEL TRAJECTORIES AND TEMPORAL TRAJECTORY FILTERING

Construction of Motion-Compensated Temporal Pixel Trajectories

In order to perform pixel-wise trajectory filtering on a video sequence, it becomes necessary to identify the individual motion of every pixel at the decoder [17]. Hence, in order to estimate the motion trajectory at the decoder the motion vectors transmitted in the bit stream are concatenated as long as the encoder estimates them valid. To this end, the concatenation of coded H.264/AVC block-based motion vectors (MVs) is now considered, which avoids the transmission of unnecessary side information to the decoder.

Concatenation of MVs

Every pixel within a motion compensated block is assumed to have the exact motion indicated by the associated motion vector, which yields two dense motion vector fields. The following description uses hierarchical B-frames as the underlying coding structure.

Let $(dx_{i,0}, dy_{i,0})^T$ and $(dx_{i,1}, dy_{i,1})^T$ shall denote the MVs for a certain block in a bidirectional predicted frame i for reference lists 0 and 1, respectively. In this context, the translational block motion is assumed to be identical to the motion of every pixel within that block, which yields the two MV fields $(dxi,_0(x, y), dyi,_0(x, y))^T$ and $(dxi,_1(x, y), dyi,_1(x, y))^T$. Frames referenced by these vectors shall be indicated by $\text{ref}_{i,0}(x, y)$ and $\text{ref}_{i,1}(x, y)$. Starting with an arbitrary pixel $(x_0, y_0)^T$ in frame i of a video sequence, the two possible new locations for a pixel reference in frames $\text{ref}_{i,0}(x, y)$ and $\text{ref}_{i,1}(x, y)$ are then given by [8][17]:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} dx_{i,0}(|x_0|,|y_0|) \\ dy_{i,0}(|x_0|,|y_0|) \end{pmatrix}$$
$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} dx_{i,1}(|x_0|,|y_0|) \\ dy_{i,1}(|x_0|,|y_0|) \end{pmatrix} \quad (5)$$

Through the concatenation of several MVs, a potential trajectory for every single pixel can now be formed. The next two locations derived from $(x_1, y_1)^T$ are, for example

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} dx_{rf,0}(|x_1|,|y_1|) \\ dy_{rf,0}(|x_1|,|y_1|) \end{pmatrix}$$
$$\begin{pmatrix} x_6 \\ y_6 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} dx_{rb,1}(|x_1|,|y_1|) \\ dy_{rb,1}(|x_1|,|y_1|) \end{pmatrix} \quad (6)$$

with
$\text{rf} = \text{ref}_{i,0}(x_1, y_1)$, $\text{rb} = \text{ref}_{i,1}(x_1, y_1)$.

This relationship and the resulting trajectory structure for an exemplary B-frame of the lowest hierarchy order are shown in Fig. 3. Contrary to the original idea of a trajectory, which associates every pixel with a single motion path through the sequence, the trajectory now branches into two individual trajectories at each B-frame with an IBBB coding structure. It becomes necessary to distinguish between those motion vectors that describe the true motion of a pixel and those that have purely been chosen due to RD-optimization [8].

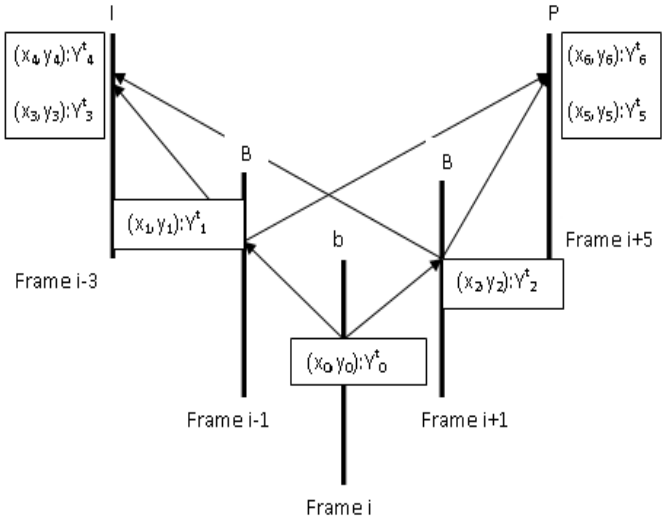


Fig. 3. Starting with a B-frame, the trajectory for each pixel is computed through the concatenation of MVs yielding here a total of seven luminance samples along the trajectory.

ISSN 2349-4506
Impact Factor: 2.545
Global Journal of Engineering Science and Research Management

Note that only some hierarchical B-frames out of a GoP of size 8 are depicted for simplification. Every one of these pixel locations yields a different luminance sample $Y^t_0,\ldots,Y^t_{N-1}$ which can potentially be used to compute a mean value Y_{opt}. The originally reconstructed pixel Y^t_0 can then be replaced by Y_{opt} to achieve noise reduction. However, not necessarily do all predicted image locations actually describe the true motion of a pixel. As in the case of frame $i-3$ in Fig. 3, only one of several pixels within an image can actually be part of the trajectory. Therefore, it becomes necessary to distinguish between MVs that do correctly describe a pixel's true motion and those that do not. Criteria that enable the decoder to perform this decision will be discussed in the subsequent section.

Distinguishing between True and False Trajectories

To enable the codec in order to select MVs that correctly describe a pixel's trajectory, three criteria have been developed that together control the filtering process with associated thresholds that are conveyed as side information in the bit stream [8][17].

1) *Absolute Error along the Trajectory:* A sudden change between consecutive luminance samples Y^t_{i+1} and Y^t_i is one the indicator for a falsely predicted trajectory. Such a change might for instance be caused by moving foreground objects covering part of the background or it could be a lighting change during the video sequence. In this case, the trajectory might still be correctly predicted, but temporal filtering would introduce further errors. Using the deviation measure $\Delta Y_i = Y^t_{i+1} - Y^t_i$, a trajectory is acceptable only if,

$$\Delta Y_i < T_Y \qquad (7)$$

2) *Temporal Motion Consistency:* It is another property of a pixel trajectory that can be used as a reliability test. It is assumed that the individual translational motion of a pixel changes only slightly from frame to frame. In order to be able to compare MVs from different frames, these are first scaled according to the reference frames they point to

$$\left(dx_{i,0}(x,y)\right)' = \frac{dx_{i,0}(x,y)}{ref_{i,0}(x,y)-i}$$
$$\left(dy_{i,0}(x,y)\right)' = \frac{dy_{i,0}(x,y)}{ref_{i,0}(x,y)-i} \qquad (8)$$

Hence, the trajectory for a given pixel is only continued if the Euclidean distance between the scaled version of the MV $(dx_i, dy_i)^T$ pointing to the current location of the trajectory and the scaled versions of the vectors $(dx_{r0,0}, dy_{r0,0})^T$ and $(dx_{r0,1}, dy_{r0,1})^T$ pointing to the reference frames for the current location is smaller than a given threshold:

$$\sqrt{((dx_{r0,0})'-(dx_i)')^2 + ((dy_{r0,0})'-(dy_i)')^2} < T_{TC} \qquad (9)$$

where the apostrophe marks that the scaled versions of the original MVs are used.

3) *Spatial Motion Consistency:* Another indicator for a motion vector that does not describe the true motion of a pixel can be found by comparing it with its neighboring vectors. Even at object boundaries, spatially adjacent MVs on the 4 ×4 block level should ideally be similar. If one vector differs significantly from its neighbors, it is assumed that a false MV has been used.

The filtering is subsequently only continued along the trajectory when the block-vote metric for the current pixel satisfies

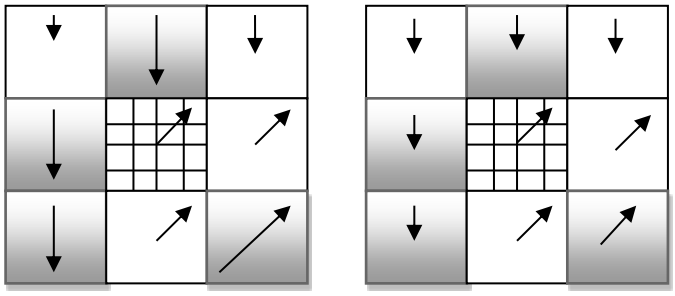
$$BV_{i,0}(x,y) \le 4 - T_{SC} \qquad (10)$$

for a given threshold T_{SC}.

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

The block-vote metric, $BV_{i,0}(x, y)$ [17] now gives the number of neighboring MVs for the 4×4 blocks surrounding the trajectory's current location (x, y), whose x or y-components differ from the current MV. Fig. 4 illustrates the principle of the block-vote metric in combination with the scaling of MVs.



(a) *(b)*

Fig. 4. (a) MVs for the 4 ×4-blocks surrounding the trajectories current location marked in black. All blocks with a gray background use a reference frame with a greater temporal distance relative to the current frame than the reference frames for the other blocks. (b) Scaled MVs. The resulting number of neighboring blocks with a similar motion (i.e., the block-vote metric) is 3.

The two thresholds T_Y and T_{TC} could theoretically take on arbitrary values. In order to reduce the time needed for parameter optimization, the two thresholds T_Y and T_{TC} are restricted to values between 0 and 7 and the appropriate range of values for the threshold T_{SC} is $0 \leq T_{SC} < 4$[17].

Temporal Trajectory Filtering (TTF)

In statistical signal theory, it is a well-known fact that, if several noisy versions of a signal are available, noise reduction can be achieved by averaging all versions of the signal. Important applications of such noise reduction concepts include microphone arrays and antenna array systems [15]. In the case of time-dependent signals such as audio, speech, or video, temporal or spatial alignment, respectively, need to be performed prior to filtering. The underlying image model for the TTF can be described as follows. For simplification, it is initially assumed that the image content of a video sequence is only subject to translational motion.

Let $Y_i(x,y)$, $U_i(x,y)$, and $V_i(x,y)$ shall denote the luminance and chrominance components of the i^{th} frame of a video sequence in display order. However, for simplicity in this paper we consider only the luminance component of the frame. In this case, for every image point $(x_0,y_0)^T$ with a luminance value of $Y_j(x_0,y_0)$ in a frame j, its location in N-1 previous frames can also be identified. These shall be denoted by $(x_i,y_i)^T$; $0 < i < N$ with the associated luminance samples $Y_{j-i}(x_i,y_i)$. Even if the motion of every pixel is known, $\hat{Y}_{j-i}(x_i, y_i)$; $0 \leq i < N$ in the decoded sequence will not be the same as the original luma sample $Y_j(x_0,y_0)^T$ due to the noise introduced during the encoding process. However, the pixel amplitude remains identical, $Y_j(x_0, y_0) = Y_{j-i}(x_i, y_i)$. Each of these shall be distorted by additive white noise n_i with variance σ^2_N as follows [17]:

$$\hat{Y}_{j-i}(x_i, y_i) = Y_{j-i}(x_i, y_i) + n_i \qquad (11)$$

The autocorrelation function of such a noise term is given by:

$$R_{NN}(k,l) = E[n_k n_l] = \begin{cases} \sigma^2{}_N, & \text{if } k = l \\ 0, & \text{else} \end{cases} \qquad (12)$$

According to [18], the noise variance in a reconstructed frame is given by

$$\sigma^2{}_i = \frac{Q^2{}_{step}}{3} \quad \text{with } Q_{step} = 0.625.2^{\frac{QP_i}{6}} \qquad (13)$$

for a given quantization parameter QP_i.

So as to achieve noise reduction in reconstructed frames, the TTF calculates the mean of the samples along a pixel's trajectory and replaces $\hat{Y}_{j-i}(x_i, y_i)$ with a filtered version of pixel $\hat{Y}_j(x_0, y_0)$ as [17]:

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

$$Y_{opt,j}(x_0,y_0) = \frac{1}{N}\sum_{i=0}^{N-1}\hat{Y}_{j-i}(x_i,y_i)$$
$$= \frac{1}{N}\sum_{i=0}^{N-1}Y_{j-i}(x_i,y_i) + \frac{1}{N}\sum_{i=0}^{N-1}n_i \qquad (14)$$
$$= \frac{1}{N}\sum_{i=0}^{N-1}Y_{j}(x_0,y_0) + \frac{1}{N}\sum_{i=0}^{N-1}n_i$$

The noise variance σ^2_{opt} of the filtered pixel is:

$$\sigma^2{}_{opt} = \frac{1}{N^2}E\left[\sum_{i=0}^{N-1}n_i\sum_{k=0}^{N-1}n_k\right] \qquad (15)$$
$$= \frac{1}{N^2}\sum_{i=0}^{N-1}\sigma^2{}_N = \frac{\sigma^2{}_N}{N}$$

Equation (15) shows that for white Gaussian noise, averaging in the temporal chronological direction over N frames will reduce the variance by a factor of N and median filtering will reduce by a factor of only 2N/Π. However, both filtering techniques in the temporal direction will degrade (blur) moving objects [9]. To reduce this degrading effect, we recommend estimating the direction of motion at each pixel and then doing the filtering along that direction.

ENCODER AND DECODER DESIGN

In general, the new in-loop filter could be applied both before and after the H.264/AVC deblocking filter. However, the filter performs better when utilized before the deblocking filter [17] to have a good spatial motion consistency. The resulting modified encoder is shown in Fig. 5. For the H.264/AVC codec, eight past reconstructed, filtered frames are stored at both encoder and decoder in a simple queue. F'_i and F'_{i-1} are the current reconstructed picture and its predecessor stored in the frame store. To allow for trajectories over at least eight past and/or future frames, the simple queue model needs to be extended when hierarchical B-frames are used. The TTF's reconstructed reference frames for trajectory formation are extracted before the deblocking filter is applied. Frames filtered by the TTF are then processed by deblocking filter. Afterwards, the most recent frame in the TTF's buffer is updated.

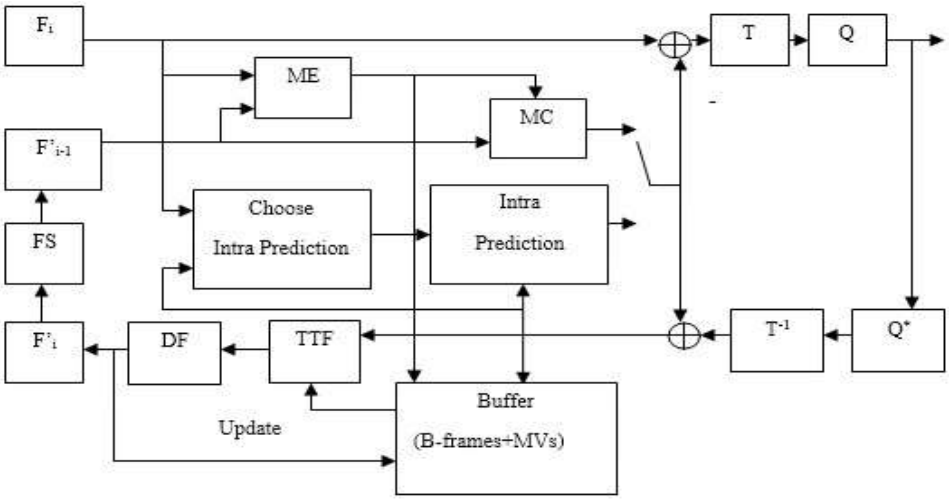


Fig.5. Proposed filter is integrated into the local decoder loop at the encoder before the deblocking filter.

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

EXPERIMENTAL RESULTS AND DISCUSSION

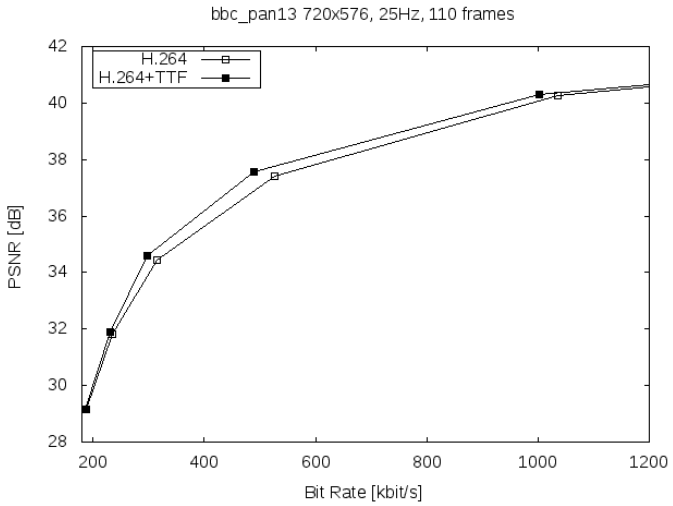
Objective Quality

The motion-compensated temporal trajectory in-loop filter has been implemented in Matlab and integrated into the current state-of-the-art codec H.264/AVC [18]. As described in Section 3, both the DF and the TTF with hierarchical B-frames have been tested. All the test sequences have been encoded using H.264/AVC with and without the additional in-loop filter. The resulting BD-rates and peak signal-to-noise ratio (PSNR) values were compared using the Bjøntegaard metric [19], which is rate-complexity-distortion evaluation for hybrid video coding. The respective Bjøntegaard delta (BD)-PSNR values and BD-rates for QPs 30 are given in Table 1.

Table 1. BD-rate and PSNR-gain results

	Test Sequences						
	All-stars (football)	Basketball	BBC_PAN13	Desert	Traffic	RaceHorses	Cactus
Resolution	704x576	1024x576	720x576	720x576	2560x1600	832x480	1920x1080
Frames/sec	25	25	25	25	30	24	50
DBF PSNR	34.28 dB @260.66kBit/s	31.85 dB @994.68kBit/s	37.40 dB @526.37kBit/s	32.52 dB @308.76kBit/s	36.24 dB @5753.49kBit/s	34.48 dB @2578.32kBit/s	32.56 dB @3507.46kBit/s
DBF+TTF PSNR	34.35 dB @257.51kBit/s	31.95 dB @971.31kBit/s	37.56 dB @488.39kBit/s	32.74 dB @289.28kBit/s	36.33 dB @5583.72kBit/s	34.52 dB @2555.40kBit/s	32.56 dB @3499.85kBit/s
BD-rate	-2.49%	-2.76%	-7.62%	-9.13%	-1.73%	-1.73%	-0.12%
PSNR-gain	0.09dB	0.11dB	0.42dB	0.44dB	0.07dB	0.07dB	0.00dB

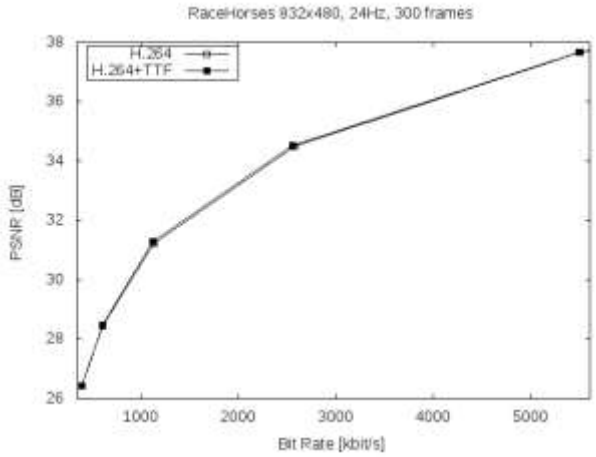
The RD-curves in Fig. 6(a) and (b) illustrate the results reported in Table 1 in more detail. In particular, good results have been achieved for both high and low-resolution sequences. Even for sequences with large moving foreground objects, such as RaceHorses and football sequences, gains are achieved. The general behavior of the temporal trajectory filter is shown by Fig. 6(a), where significant quality improvement for the test sequence is present. For the *bbc_pan13* sequence, the TTF provides objective quality improvement for all depicted bit rates, while for the *RaceHorse* sequence; the gain is restricted to the medium bit rate range.



(a)

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management



(b)

Fig. 6. Exemplary RD-curves for H.264 debloking filter and motion compensated temporal trajectory filtering. (a) BBC_pan13, (b) RaceHorse.

Subjective Quality

Apart from improving the objective quality of the decoded video, the additional in-loop filter also improves the subjective quality. As outlined in Section 3, the main purpose of the TTF is to reduce the impact of block artifacts on the performance of H.264/AVC DF. These are expected to occur especially at the boundaries of moving objects in a video sequence and at high QP-values. Fig. 7 shows frame 290 of the decoded *RaceHorse* sequence both with and without the application of the TTF filter. A combination with the Wiener-based ALF has also been tested in [17] indicating that both filters in combination can potentially provide an even higher gain.

(a)

(b)

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

(c)

Fig. 7. Enlarged part of frame 290 from the decoded RaceHorse sequence. (a) Original frame, (b) H.264/AVC (DF), (c) DF+ TTF.

CONCLUSION

This paper has presented a technique for improving the performance of deblocking filter using temporal trajectory filtering by the incorporation of motion estimation and compensation. The method gives the possibility of reducing noise in moving areas without affecting image detail; this is not possible in noise reducers which do not perform motion estimation. Simulation results have shown that this technique gives a quality comparable to or better than that of the H.264/AVC deblocking filter using previous frame prediction. When used in conjunction with an inter-frame predictive coder, that already incorporate motion estimation, the improvement in quality coupled with the increased efficiency of the coder make this an attractive approach to noise reduction. Since the picture quality is always improved, all three thresholds described in this paper appear to be good indicators for correctly predicted trajectories. Future work therefore will focus on integrating RD-optimization scheme into the TTF concept and on the computational simplification of both encoder and decoder.

REFERENCES

1. DUBOIS, E.—SABRI, S.: Noise Reduction in Image Sequences Using Motion-Compensated Temporal Filtering, IEEE Transactions on Commn **32** No. 7 (1984), 826-831.
2. LIST, P.—JOCH, A.—LAINEMA, J.—BJØNTEGAARD, G.—KARCZEWICZ, M.: Adaptive deblocking filter, IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) **13** No. 7 (2003), 614–619.
3. MCCANN, K.—WIEGAND, T.—BROSS, B.—HAN, W.-J.—OHM, J.-R.—RIDGE, J.—SEKIGUCHI, S.—SULLIVAN, G. J.: HEVC draft and test model editing, document JCTVC-D500-r1.doc, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 (2011).
4. FU, C.-M.—CHEN, C.-Y.—HUANG, Y.-W.—LEI, S.: Sample adaptive offset for HEVC, IEEE 13th Intl. Workshop on Multimedia Signal Processing (MMSP) (2011), 1–5.
5. WITTMANN, S.—WEDI, T.: Transmission of post-filter hints for video coding schemes, IEEE International Conference on Image Processing (ICIP) **1** (2007), I-81 - I-84.
6. CHUJOH, T.—WADA, N.—WATANABE, T.—YASUDA, G.—YAMAKAGE, T.: Specification and experimental results of quadtree-based adaptive loop filter, document VCEG-AK22, ITU-T SG16/Q.6 VCEG (2009).
7. TAN, T. K.—SULLIVAN, G.—WEDI, T.: Recommended simulation common conditions for coding efficiency experiments, document VCEG-AA10, ITU-T SG16/Q.6 VCEG (2005).
8. ESCHE, M.—KRUTZ, A.—GLANTZ, A.—SIKORA, T.: A novel in loop filter for video compression based on temporal pixel trajectories, 28th Picture Coding Symposium, PCS2010, , Nagoya, Japan, (2010), 514-517
9. PAQUIN, R.—DUBOIS, E.: A spatio-temporal gradient method for estimating the displacement field in time-varying imagery, Computer vision, graphics, and image processing **21** No.2 (1983), 205-221.
10. TISTARELLI, M.: Computation of optical flow and its derivatives from local differential constraints, Proceedings, International Symposium on Computer Vision (1995), 19 - 24.

11. HUANG, T.S.—HSU, Y.P.: Image sequence enhancement, Chapter: Image Sequence Analysis of the series Springer Series in Information Sciences **5** (1981), 289-309.
12. OHM, J.-R.: Three-dimensional subband coding with motion compensation, IEEE Trans. Image Process. **3** No. 5 (1994), 559–571.
13. CHENG, C.C.—CHEN, C.Y.—CHEN, Y.H.—CHEN, L.G.: Analysis and VLSI Architecture of Update Step in Motion-Compensated Temporal Filtering, Proceedings IEEE International Symposium on Circuits and Systems (ISCAS) (2006), 5343-5346.
14. VO, D. T.—NGUYEN, T. Q.: Optimal motion compensated spatiotemporal filter for quality enhancement of H.264/AVC compressed video sequences, 16th IEEE International Conference on Image Processing (2009), 3173–3176.
15. GLANTZ, A.—KRUTZ, A.—HALLER, M.—SIKORA, T.: Video coding using global motion temporal filtering, 16th IEEE International Conference on Image Processing (2009), 1053–1056.
16. DENNIS, T. J.: Nonlinear temporal filter for television picture noise reduction, IEE Proceedings G Electronic Circuits and Systems **127** (1980), 52-56.
17. ESCHE, M.—GLANTZ, A.—KRUTZ, A.—SIKORA, T.: Adaptive Temporal Trajectory Filtering for Video Compression, IEEE Transactions On Circuits And Systems For Video Technology **22**, No. 5 (2012), 659-670.
18. WIEGAND, T.—SULLIVAN, G.—BJØNTEGAARD, G.—LUTHRA, A.: Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. **13** No. 7 (2003), 560–576.
19. BJØNTEGAARD, G.: Calculation of average PSNR differences between RDcurves, document VCEG-M33, ITU-T SG16/Q.6 VCEG, (2001).