Motion Compensated Three-Dimensional Frequency Selective Extrapolation for Improved Error Concealment in Video Communication

Jürgen Seiler*, André Kaup
Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, Cauerstr. 7, 91058 Erlangen, Germany

Abstract
During transmission of video data over error-prone channels the risk of getting severe image distortions due to transmission errors is ubiquitous. To deal with image distortions at decoder side, error concealment is applied. This article presents Motion Compensated Three-Dimensional Frequency Selective Extrapolation, a novel spatio-temporal error concealment algorithm. The algorithm uses fractional-pel motion estimation and compensation as initial step, being followed by the generation of a model of the distorted signal. The model generation is conducted by an enhanced version of Three-Dimensional Frequency Selective Extrapolation, an existing error concealment algorithm. Compared to this existent algorithm, the proposed one yields an improvement in concealment quality of up to 1.64 dB PSNR. Altogether, the incorporation of motion compensation and the improved model generation extends the already high extrapolation quality of the underlying Frequency Selective Extrapolation, resulting in a gain of more than 3 dB compared to other well-known error concealment algorithms.

Keywords: Signal Extrapolation, Error Concealment, Video Communication

*The work of J. Seiler and A. Kaup was partly supported by the German Research Foundation Deutsche Forschungsgemeinschaft (DFG) under contract number KA 926/1-2.
*Corresponding author, phone: +49 9131 85 27103, fax: +49 9131 85 28849
Email addresses: seiler@lnt.de (Jürgen Seiler), kaup@lnt.de (André Kaup)

Preprint submitted to Journal of Visual Communication and Image Representation April 28, 2022
1. Introduction

In the past years the computing power of multimedia devices has been steadily increased. In combination with the growing transport capacity for mobile transmission and transmission over the internet, the ability to record and play digital video data has become more and more popular. But by transporting digital video data over wireless channels or the internet, the risk of occurring transmission errors is ubiquitous. On wireless channels the errors may be caused e. g. by fading or the loss of synchronization. By using the packetized data transport of the internet, the main causes for transmission errors are delayed or lost data packets. No matter what actually caused the transmission error, the decoded video sequence becomes heavily distorted. Due to the transmission errors, parts of one or several video frames cannot be displayed, or only with severe distortion. The errors become even worse if a hybrid video codec as e. g. the H.264/AVC [1] is used, since the errors propagate to following frames due to the predictive structure of such a video codec. A hybrid video codec uses already transmitted parts of the sequence for predicting the block being coded. Thus, the decoder can predict the block in the same manner as the encoder. But in the case of a distortion, the decoder has to form the prediction based on the distorted signal parts and therewith propagates the errors to future frames.

In order to cope with this, as shown in [2] and [3], modern hybrid video codecs use two strategies. On the one hand side, this is error resilience which aims at protecting the coded video data bitstream against transmission errors. Unfortunately, the protection is only possible at the expense of an increased data rate and therewith a decreased coding efficiency. On the other hand side, the second strategy is error concealment which takes effect in the case that a transmission error occurs and error resilience is not used or fails to correct the error. Error concealment aims at concealing the distortion caused by the transmission errors at the decoder side by extrapolating the signal from correctly received regions into the areas where the signal is distorted. So, the goal is to reconstruct the original signal for generating a pleasant image for presentation despite the error and for reducing error propagation. Since error concealment is not part of standardization, one can freely decide which concealment algorithm to use at the decoder, depending on its computational demands and memory requirements.

Within the scope of this article we focus on error concealment and present a powerful spatio-temporal error concealment algorithm, the Motion Compensated Three-Dimensional Frequency Selective Extrapolation. This algorithm uses motion estimation and compensation as initial part of the extrapolation process which is then followed by a model generation step.

In the subsequent sections, first an overview of different error concealment algorithms is given. Afterwards, we outline the novel algorithm in detail, show how the model generation part can be significantly enhanced and especially point out how the novel algorithm can use motion estimation and compensation with fractional-pel accuracy. Finally, we prove the abilities of the algorithm by a comprehensive analysis of the concealment quality for coded video data.
We also show how, in addition to the novel algorithm, other common temporal error concealment algorithms can be improved by using fractional-pel motion estimation and compensation.

2. Previous Work on Spatial, Temporal and Spatio-Temporal Error Concealment

The first group of error concealment algorithms to mention is the group that only uses spatial information from the currently considered frame for concealing the distorted areas. One member of this group is the Maximally Smooth Recovery algorithm proposed by Wang et. al. [4]. This algorithm aims at extrapolating distorted areas in such a way that a smooth transition to the correctly received neighboring samples results. Another algorithm, introduced by Sun and Kwok [5], makes use of projection onto convex sets for concealing the distortion. There, the block to be reconstructed is iteratively manipulated in the spatial and in the frequency domain in order to form an extrapolated signal that fulfills some spatial as well as some frequency constraints. Alkachouh and Bellanger [6] proposed another spatial error concealment algorithm working in the DCT domain. This algorithm recovers the lowpass DCT coefficients of the distorted block by means of the correctly received adjacent samples. Another spatial error concealment algorithm is the sequential error concealment proposed by Li and Orchard [7]. There, the distorted block iteratively is reconstructed and superimposed from different directions. A different approach is the Two-Dimensional Frequency Selective Extrapolation from [8] which iteratively generates a model of the distorted signal as a weighted superposition of two-dimensional basis functions. Besides this small selection of spatial error concealment algorithms many more exist. Some additional algorithms can be found in [9] and [10].

According to [9, 10, 11], due to the usually very high temporal correlations within one video sequence, using temporal error concealment algorithms in general is superior to using spatial ones. The group of temporal error concealment algorithms makes use of signal parts in frames that have been transmitted and decoded prior to the distorted frames. All algorithms of this group have in common that they seek an area in a previous frame to replace the distorted one in the current frame. Several of these algorithms are introduced in [9, 10, 11]. Out of this group we especially focus on the Extended Boundary Matching Algorithm (EBMA) from Lam et. al. [12] and the Decoder Motion Vector Estimation (DMVE) proposed by Zhang et. al. [13] since these two algorithms are widely used and will later serve for comparison. Both algorithms aim at determining the motion of the sequence in order to locate a shifted area in a previous frame which then is used for replacing the distorted area. For this, EBMA selects the area that minimizes the boundary error between the correctly received adjacent samples in the distorted current frame and the margin of the candidate area from the previous frame. If DMVE is used, an area of correctly received samples around the distortion is selected. Then, this area is compared to shifted versions of itself in the previous frame. The estimated motion corresponds to the area minimizing the error. After this, the distorted area is replaced by the
one from the previous frame shifted according to the estimated motion. In the case of bi-temporal prediction where previous as well as future frames can be available, the best matching area can also be searched in a future frame. Both algorithms can use a full search as well as a restricted search for determining the candidates. Thereby, the restricted search only uses a subset of candidate motion vectors in order to select the best one among them. This subset can be obtained e. g. by regarding the motion vectors of the correctly received neighboring blocks.

The third group of error concealment algorithms, the spatio-temporal error concealment algorithms, exploits spatial as well as temporal information and therewith is superior to spatial and temporal error concealment according to [9]. The first algorithm out of this group to mention is the Spatio-Temporal Fading Scheme from Friebe and Kaup [14] and its extension to B-Frames [15]. This algorithm first performs a temporal and a spatial extrapolation and then conducts a weighted superposition of them. For this, the weights for the spatial and the temporal extrapolation are determined by using the boundary error. This algorithm is improved by Hwang et. al., leading to the Improved Fading Scheme (IFS) [16]. There, a superposition of DMVE and EBMA is used for temporal extrapolation and a superposition of stripe patch repetition and bi-linear interpolation for spatial extrapolation. The Spatio-Temporal Boundary Matching Algorithm [17] from Chen et. al. follows a different approach. This algorithm exploits temporal and spatial smoothness of a signal to recover a motion vector and therewith is an extension of the original EBMA. An improvement to the original DMVE is achieved by Multi-Hypothesis Error Concealment (MH-EC) [18] from Song et. al. This algorithm determines several candidate blocks that can be used for concealing the distortion and superimposes them. With that, the idea of Multihypothesis Motion Estimation from Flierl and Girod [19] is applied to error concealment. The Spatio-Temporal Error Concealment with Optimized Mode Selection from Belfiore et. al. [20] is another algorithm out of this group and is able to effectively switch between spatial and temporal extrapolation and a combination of them. A similar approach is proposed by Hadar et. al., regarding their Hybrid Error Concealment [21], where the switch between the different initial signal extrapolations is performed differently. Another spatio-temporal error concealment algorithm is the Boundary Matching Algorithm and Mesh-Based Warping from Atzori et. al. [22]. There, first a preliminary temporal extrapolation is obtained by using the Boundary Matching Algorithm, which is spatially warped to fit the correctly received neighborhood afterwards. A completely different approach is the spatio-temporal concealment by Three-Dimensional Frequency Selective Extrapolation (3D-FSE) proposed by Meisinger and Kaup in [23] which is the extension of the algorithm from [8] into three dimensions. This algorithm aims at generating a joint three-dimensional model of the signal suffering from distortion.

Unfortunately, 3D-FSE is only able to compensate small motion effectively. In the case of large motion, the model generation is performed only suboptimally and the generated model may not fit the original signal well. In order to cope with this shortcoming, we propose the subsequently described Motion
Compensated Three-Dimensional Frequency Selective Extrapolation. The algorithm is an enhancement to 3D-FSE and uses an explicit estimation of the motion in a sequence and takes this into account for the model generation.

3. Motion Compensated Three-Dimensional Frequency Selective Extrapolation

Before actually presenting Motion Compensated Three-Dimensional Frequency Selective Extrapolation (MC-FSE) in detail, we will briefly review the existing 3D-FSE from [23] in the next subsection. So, we can point out its shortcoming in scenes with motion and therewith we can show the necessity of MC-FSE. In addition to that, we will also introduce an effective modification to the original model generation algorithm which is also a part of the later outlined MC-FSE.

3.1. Three-Dimensional Frequency Selective Extrapolation

For formulating 3D-FSE, and later the MC-FSE algorithm, the following scenario is regarded: we consider the video sequence $v(x, y, t)$ which is shown in Figures 1 and 3 top and is depicted by spatial coordinates $x$ and $y$ and temporal coordinate $t$. The spatial size of the sequence is $X \times Y$. Without loss of generality, an isolated block loss of size $L_x \times L_y$ samples occurs in the frame $t = \tau$ at position $(x_0, y_0)$. The concealment of an isolated block loss is considered only for a comprehensible description. In general, the areas to conceal can be arbitrarily shaped. In such cases, as described in [8], the area to conceal is divided into blocks and the individual blocks are concealed successively. In the course of this procedure, already concealed block are used for supporting the concealment of the next ones.

For concealing the loss, 3D-FSE utilizes correctly received samples from the current and from previous frames and, if available, future frames. The number
of involved previous frames is depicted by $N_p$, of future frames by $N_f$. As mentioned before, 3D-FSE aims at generating a model of the signal suffering from the distortion. For this, a data volume, the so called extrapolation volume $L$, centered by the loss, is cut out of the video sequence. This volume consists of the loss of size $L_x \times L_y$ samples, surrounded by a border of $B$ samples width from the frame $t = \tau$. In addition to this, from the involved previous and future frames the samples at the same spatial positions are taken. The extrapolation volume $L$ is depicted by coordinates $m$, $n$ and $p$ and is of size $M \times N \times P$ with

$$
M = L_x + 2B 
$$  \hspace{1cm} (1)

$$
N = L_y + 2B 
$$  \hspace{1cm} (2)

$$
P = N_p + N_f + 1. 
$$  \hspace{1cm} (3)

The signal in volume $L$ is depicted by $f [m, n, p]$ and emanates from $v [x, y, t]$ according to

$$
f [m, n, p] = v [x_0 - B + m, y_0 - B + n, \tau - N_p + p] . 
$$  \hspace{1cm} (4)

Further, volume $L$ can be split into two parts, region $B$ containing the lost samples and support volume $A$, subsuming all correctly received samples used for model generation. The relation between video sequence $v [x, y, t]$ and signal $f [m, n, p]$ in volume $L$ can also be seen in Fig. 1.

Then, a model of the signal is generated in this volume. This model is obtained by iteratively approximating the correctly received signal parts in $A$, but as the model is defined over the complete volume $L$, an estimate of the lost block $B$ can be achieved. Subsequently, the algorithm is outlined as sparse approximation problem. Sparsity-based algorithms are suited well for solving underdetermined problems, to which the extrapolation problem belongs. As shown in [24], these algorithms are able to cover important signal characteristics even if the underlying problem is underdetermined. The Frequency Selective Extrapolation belongs to the group of greedy sparse approximation algorithms as described in [25] and [26]. Another approximation algorithm out of this group is Matching Pursuit, originally proposed by Mallat and Zhang [27]. These iterative greedy approximation algorithms can solve sparse problems efficiently and are much more suited for the model generation than a direct solution of the sparse problem using the $L_0$-norm since this is a NP-hard problem as shown in [28].

3D-FSE now aims at generating the parametric model

$$
g [m, n, p] = \sum_{k \in \mathcal{K}} \hat{c}_k \varphi_k [m, n, p] 
$$  \hspace{1cm} (5)

over the complete extrapolation volume $L$. The model is a weighted superposition of the mutually orthogonal basis functions $\varphi_k [m, n, p]$, and $\hat{c}_k$ denotes the individual weights. The set $\mathcal{K}$ contains the indices of all basis functions used for model generation. The model generation algorithm on the one hand side has to determine which basis functions to use for the model and on the other hand
side has to calculate the corresponding weights. These two tasks are carried out in an iterative manner and, in doing so, one basis function is selected and the corresponding weight is calculated in every iteration.

For generating the signal model by 3D-FSE, the three-dimensional signals in extrapolation volume $L$ are considered as vectors with the reversible mapping

$$[x]_{m+M \cdot n+M \cdot N \cdot p} = x[m, n, p]$$

(6)

with

$$m = 0, \ldots, M - 1$$

$$n = 0, \ldots, N - 1$$

$$p = 0, \ldots, P - 1.$$

This mapping produces the vector $x$ for any arbitrary signal $x[m, n, p]$. Expression $[x]_i$ is used to indicate the $i$-th element of the vector $x$. According to this, signal $f[m, n, p]$ is represented by $f$, model $g[m, n, p]$ by $g$ and the basis functions $\varphi_k[m, n, p]$ by basis vectors $\varphi_k$. In addition to this, the basis vectors are subsumed in the matrix

$$\Phi = \left( \varphi_0, \varphi_1, \ldots, \varphi_{|L|-1} \right).$$

(7)

Thus, the model vector $g$ results from

$$g = \Phi \hat{c}$$

(8)

with the coefficient vector $\hat{c}$ to be determined in the following. The coefficient vector $\hat{c}$ is built up iteratively in such a way that the approximation error between the model and the undistorted signal parts is reduced in every iteration. For this, the residual vector $r^{(0)}$ before the first iteration is set to

$$r^{(0)} = f.$$

(9)

and the coefficient vector $\hat{c}$ is set to 0.

At the beginning of each iteration, here the $\nu$-th one is regarded, a weighted projection of the last iteration’s approximation residual $r^{(\nu-1)}$ onto all possible basis vectors is performed. This yields the projection vector $p^{(\nu)}$ which can be calculated as follows:

$$p^{(\nu)} = \left( r^{(\nu-1)} \right)^T W \Phi \left( \text{diag} \left( \Phi^T W \Phi \right) \right)^{-1}$$

(10)

The weighting matrix $W = \text{diag} (w)$ is used to mask the lost area and to control the influence each sample has on the model generation depending on its position. $w$ and therewith $W$ emanate from the three-dimensional weighting function:

$$w[m, n, p] = \begin{cases} \beta \sqrt{(m-M-1)^2 + (n-N-1)^2 + (p-P-1)^2} & \forall (m, n, p) \in A \\ 0 & \forall (m, n, p) \in B \end{cases}$$

(11)
According to [23], the samples in the support volume are weighted by an isotropic model with decay factor $\hat{\rho}$. With that, samples get lower weight and consequently less influence on the model generation with increasing distance to the lost block. In the case that consecutive block losses are considered, the already concealed blocks are used for concealment as well. For this purpose the region corresponding to the already concealed blocks is further weighted by the reliability coefficient $\delta$ which is smaller than 1.

After having determined projection vector $p^{(v)}$, one basis vector has to be selected to be added to the model in this iteration. For this, the vector is selected that minimizes the weighted distance between the residual $r^{(v-1)}$ and the weighted projection onto the corresponding basis vector. So the index $u$ of the basis vector to be added can be determined by

$$ u = \text{argmin}_k \left( \frac{\left( \left( r^{(v-1)} - \left[ p^{(v)} \right]_k \varphi_k \right)^\top \cdot W \cdot \left( r^{(v-1)} - \left[ p^{(v)} \right]_k \varphi_k \right) \right)}{\varphi_k^\top \cdot W \cdot \varphi_k} \right) $$  \hspace{1cm} (12)

$$ = \text{argmax}_k \left( \frac{\left( \left[ p^{(v)} \right]_k \varphi_k \right)^\top \cdot W \cdot \varphi_k} \right) $$  \hspace{1cm} (13)

$$ = \text{argmax}_k \left[ \left[ p^{(v)} \right]_k \text{diag} \left( \left[ p^{(v)} \right]_k \right) \cdot \text{diag} \left( \Phi^\top \cdot W \cdot \Phi \right) \right] $$ \hspace{1cm} (14)

Here, the fact is exploited that the vector $\left( r^{(v-1)} - \left[ p^{(v)} \right]_k \varphi_k \right)$ is orthogonal to $\varphi_k$ with respect to the weighted scalar product

$$ \left( r^{(v-1)} - \left[ p^{(v)} \right]_k \varphi_k \right)^\top \cdot W \cdot \varphi_k = 0. $$ \hspace{1cm} (15)

In the next step, the coefficient vector $\hat{c}$ has to be updated. The update should cover the original portion the selected basis vector $\varphi_u$ has of the residual vector $r^{(v-1)}$. Unfortunately, due to the loss, the original portion could not be recovered exactly. However, the weighted projection of $r^{(v-1)}$ onto the basis vector $\varphi_u$ leads to a good initial estimate for the original portion and could be used for updating the coefficient vector. The original 3D-FSE algorithm from [23] directly uses the result from the weighted projection as estimate for the weight of the corresponding basis vector. But by using the result from the weighted projection directly, the update suffers from orthogonality deficiency, as has been shown in detail in [29] for the two-dimensional case. The problem is, that although the basis vectors are mutually orthogonal, they are not orthogonal anymore, if the scalar product is evaluated only over support volume $\mathcal{A}$ and in combination with weighting matrix $W$. Due to this deficiency of orthogonality, the weighted projection of $r^{(v-1)}$ onto the basis vector $\varphi_u$ contains portions of basis vectors unlike $\varphi_u$ as well. In Fig. 2 this circumstance is presented for a simple example. There, the residual vector $r$ emanates from the superposition of $\varphi_1$, $\varphi_2$ and $\varphi_3$ with original weights $c_1$, $c_2$ and $c_3$. In the regarded iteration, basis vector $\varphi_1$ is determined to be added to the model. Now, the goal is to derive a good estimate for $c_1$ from the projection coefficients. Apparently, directly using $p_1$ as estimate for $c_1$ is not optimal, since
The basis vector $\varphi_1$ would become overemphasized in the model. In order to cope with this, in [29], we proposed an algorithm using all projection coefficients for compensating the orthogonality deficiency and for obtaining good estimates in the context of two-dimensional extrapolation. Unfortunately, this approach is computationally very expensive. But the analyses presented in [30] showed that it is as well sufficient to use a simple orthogonality deficiency compensation which is carried out by only adding a fraction of the projection coefficient to the coefficient vector in a certain iteration. According to [30], the utilization of this simple orthogonality deficiency compensation has two advantages over the original, uncompensated model generation. Firstly, the extrapolation quality can be significantly improved as the risk of overemphasizing a basis vector is effectively eliminated. Secondly, the original Frequency Selective Extrapolation suffers from the problem, that the best possible extrapolation quality is achieved only for a certain number of iterations, which furthermore depends on the content. If the number of performed iterations exceeds this point, the quality degrades very fast. But, if orthogonality deficiency compensation is applied, the extrapolation quality runs into saturation with an increasing number of iterations. Hence, the threat of missing the peak extrapolation quality can be avoided. Utilizing orthogonality deficiency compensation only has one small drawback: the number of iterations which is necessary for model generation has to be increased, compared to the model generation without compensation. Since the advantages of the simple orthogonality deficiency compensation exceed the solely increased number of iterations, this approach is applied to the three-dimensional model generation in the following.

With that, entry $\hat{c}^{(\nu)}_u$ of the coefficient vector in the $\nu$-th iteration corresponding to the selected basis vector $\varphi_u$ is updated as follows:

$$\hat{c}^{(\nu)}_u = \hat{c}^{(\nu-1)}_u + \gamma \left[ p^{(\nu)} \right]_u$$

(16)

The factor $\gamma$ is from the range between 0 and 1 and causes that only the mentioned fraction of the weighted projection’s result gets added to the coefficient vector. As has also been shown in [30], fortunately the exact value of $\gamma$ is not very critical. In the case, that the fraction of the basis vector is estimated too
small, the same basis vector gets selected again in a later iteration.

After the coefficient vector has been updated, the new residual

\[
r^{(\nu)} = f - \Phi \hat{c}^{(\nu)}
\]

\[
= r^{(\nu-1)} - \gamma \left[ p^{(\nu)} \right] u \varphi
\]  

is calculated. Then, the new residual \(r^{(\nu)}\) is used for further updating the model. Altogether, the above described steps of selecting one basis vector to be added and calculating the corresponding weight are repeated until a predefined number of iterations is reached.

Finally, \(g\) is retransformed to the three-dimensional model \(g[m, n, p]\). The values of \(g[m, n, p]\) corresponding to the lost samples from area \(B\) are cut out of the model and are used for replacing the lost signal parts and therewith conceal the error.

In order to indicate the utilization of orthogonality deficiency compensation and to distinguish the novel model generation from the original 3D-FSE from [23], the enhanced version is referred to as Orthogonality Deficiency Compensated Three-Dimensional Frequency Selective Extrapolation (3D-FSE-OD).

3.2. Motion Compensated Three-Dimensional Frequency Selective Extrapolation Principles

Regarding Fig. 1 closely, it becomes obvious, that the distorted object is moving and is at different positions in the individual frames. The movement from frame \(t = \tau\) to frame \(t = \tau + \kappa\) can be described by the motion vector

\[
d^{(\kappa)} = \left( x_d^{(\kappa)}, y_d^{(\kappa)} \right)
\]  

where \(x_d^{(\kappa)}\) is the displacement in horizontal and \(y_d^{(\kappa)}\) in vertical direction. As can be seen in Fig. 1 bottom, due to the motion, the fixed data volume used for

Figure 3: Top: distorted video sequence \(v[x, y, t]\) with isolated block loss in frame \(t = \tau\) at \((x_0, y_0)\). Bottom: aligned extrapolation volume \(L_a\).

Video sequence

Motion compensated and aligned extrapolation volume
the original 3D-FSE and the enhanced 3D-FSE-OD cannot be optimal for the model generation, since the signal parts that are lost in the current frame are at different positions in the previous or future frames. Even worse, it may be possible that they are not part of a fixed volume at all.

To overcome this weakness, the proposed MC-FSE aims at revoking the motion of the object in order to get an aligned volume before the model generation takes place. In doing so, Fig. 3 top shows the same video sequence as before, but there the extrapolation volume is aligned according to the motion. In Fig. 3 bottom, the aligned extrapolation volume $L_a$ is presented for this example. Obviously, such an aligned extrapolation volume is much more suited for the subsequent model generation as the different layers of the volume, corresponding to the considered previous and future frames, contain the same signal parts as the layer corresponding to the frame suffering from the loss. In addition to that, every layer contains the region corresponding to the loss at the same position and therewith the lost area can be reconstructed more precisely. The aligned extrapolation volume is also depicted by coordinates $m$, $n$, and $p$ and is of size $M \times N \times P$. Volume $L_a$ again can be split into two parts, region $B$ subsuming the lost samples and support volume $A$, subsuming the correctly received samples.

Fig. 4 shows a block diagram of the different steps of MC-FSE that are required for obtaining the aligned volume and for extrapolating the signal. Unlike [31], where only full-pel accuracy is used for motion estimation, we propose to
estimate the motion in the sequence around the lost block with fractional-pel accuracy in the first step. Since motion estimation may be inaccurate, its reliability has to be checked. If the estimation is considered to be reliable, the extrapolation volume is aligned according to the estimated fractional-pel positions before the above described 3D-FSE-OD is applied on the aligned volume for generating the model. Otherwise, the estimation is discarded and 3D-FSE-OD is applied on the fixed volume, centered by the lost block. After the model has been generated, again the samples corresponding to the lost block are cut out of the model and are used for concealing the loss.

3.3. Motion Estimation

As has been shown in [32, 33, 34], video sequences can be compressed more efficiently by using motion compensated prediction with fractional-pel accuracy. This is due to the fact that the motion in a sequence does not necessarily have to follow the artificially introduced sampling grid of the sensor. Although fractional-pel motion estimation is quite ordinary in encoding video sequences it is used only very rarely for error concealment. The algorithms proposed in [35] and [36] are two of the few examples for using fractional-pel motion estimation, but the vast majority of extrapolation algorithms only uses full-pel accuracy. Since error concealment can be regarded as a signal extrapolation task closely related to motion compensated prediction, the idea of fractional-pel motion estimation is used subsequently. For this, the undistorted previous and future frames have to be upsampled. To simplify matters, only integer scaling factors are considered. The upsampling factor is denoted by $D$. The upsampling is performed by first inserting $D - 1$ zeros between every two neighboring samples. Afterwards, these zeros are removed by interpolating the signal between the original samples. The most simple interpolation would be just using bilinear filtering. Alternatively, more sophisticated interpolation schemes like the two-stage interpolation used in H.264/AVC [1] or the adaptive interpolation proposed in [37] can be used. Subsuming all regarded frames, we get the upsampled sequence $v_u[x, y, t]$ of spatial size $(D(X - 1) + 1) \times (D(Y - 1) + 1)$. Thereby, the original full-pel samples get mapped from spatial coordinates $(x, y)$ to $(Dx, Dy)$.

After the upsampling of the reference frames, the motion in the sequence
around the lost block is estimated. Here, the limitation has to be considered, that the samples of the lost block cannot be taken into account for estimating the motion. To cope with this, the same idea is used as is utilized in DMVE for recovering the lost motion vector. There, an area around the lost block is regarded and motion estimation is performed for this area, meaning that the best matching area in the regarded frame is determined. Fig. 5 shows the relation between the so called decision area $M$ and the lost block. To estimate the motion of the object between the frame at $t = \tau$ and the upsampled frame at $t = \tau + \kappa$, all motion vectors up to a maximum displacement of $d_{\text{max}}$ are tested. For this, for every candidate motion vector $\hat{d}(\kappa) = (\hat{x}_d(\kappa), \hat{y}_d(\kappa))$ the sum of squared errors

$$E(\kappa)(\hat{d}(\kappa)) = \sum_{(x,y) \in M} (v[x, y, \tau] - v_u[Dx + \hat{x}_d(\kappa), Dy + \hat{y}_d(\kappa), \tau + \kappa])^2$$

(20)

between all samples from area $M$ in frame $t = \tau$ and the displaced version of $M$ in frame $t = \tau + \kappa$ is calculated. If a candidate vector points outside the margins of the examined frame, the regarded frame is padded respectively with the samples of its margin. $\kappa$ indicates all the frames used for concealment and is from the range $-N_p, \ldots, N_f \setminus 0$.

After the sum of squared errors has been determined for every candidate motion vector, the one is selected, that minimizes the error, resulting in

$$\hat{d}(\kappa) = \arg\min_{\hat{d}(\kappa)} E(\kappa)(\hat{d}(\kappa))$$

(21)

$$= \arg\min_{\hat{x}_d(\kappa), \hat{y}_d(\kappa) = -d_{\text{max}}, \ldots, d_{\text{max}}} E(\kappa)((\hat{x}_d(\kappa), \hat{y}_d(\kappa)))$$

(22)

Since the reliability of the estimation is checked in the next step, the estimation error

$$\hat{E}(\kappa) = E(\kappa)((\hat{d}(\kappa)))$$

(23)

belonging to the selected motion vector is stored for further usage.

These steps are performed for all considered previous and future frames. This results in a set of estimated motion vectors $\hat{d}(\kappa)$ with $\kappa = -N_p, \ldots, N_f \setminus 0$ and the estimation errors $\hat{E}(\kappa)$ corresponding to each of the vectors.

3.4. Motion Estimation Evaluation

The above mentioned reliability check for the motion estimation quality is important since motion cannot be estimated well in some cases, e. g., if non-translational movements or content changes occur. In such a case, the estimated motion vectors do not fit the real motion in the scene and an extrapolation volume $L_a$ aligned according to those vectors is less suited for the subsequent model generation than an unaligned one. In order to evaluate the estimation reliability, two thresholds are used for checking the absolute quality of the estimation.
and its homogeneity. In the case that at least one of the two thresholds is exceeded, the estimation is discarded and the model generation is performed on the unaligned, fixed volume.

For testing the absolute motion estimation quality, the maximum of the square root of the estimation errors $\sqrt{E(\kappa)}$, normalized by the number of samples in $M$ is used. Thus, the maximum mean error per pixel is calculated for the motion vectors to all regarded frames. If this value is above threshold $T_{\text{abs}}$, i. e.

$$\max_{\kappa=-N_p,...,N_f\setminus 0} \sqrt{\frac{E(\kappa)}{|M|}} > T_{\text{abs}},$$

the motion estimation is regarded as unreliable. For checking the homogeneity of the estimation, the difference between the maximum and the minimum estimation error is calculated and afterwards normalized by the mean estimation error of all frames. This quotient

$$\frac{\max_{\kappa=-N_p,...,N_f\setminus 0} \sqrt{E(\kappa)} - \min_{\kappa=-N_p,...,N_f\setminus 0} \sqrt{E(\kappa)}}{\sum_{\kappa=-N_p,...,N_f\setminus 0} \sqrt{E(\kappa)}} > T_{\text{rel}}$$

is compared to the second threshold $T_{\text{rel}}$. Again, if this threshold is exceeded, the motion estimation for this volume is decided to be unreliable and is discarded.

3.5. Volume Alignment

After the motion in the sequence has been estimated and the estimation has been evaluated, the extrapolation volume $L_a$ and therewith $f[m, n, p]$ is set up. In the case that the estimation is regarded as unreliable, the extrapolation volume consists of the samples from frame $t = \tau$ belonging to the lost block, a border of $B$ samples width surrounding the loss, and the samples at the same spatial positions in the considered previous and future frames. Thus, signal $f[m, n, p]$ in this volume emanates from the video sequence according to:

$$f[m, n, p] = v[x_0 - B + m, y_0 - B + n, \tau - N_p + p].$$

If the estimation is classified as reliable, from the frame at $t = \tau$ the same samples as above are taken for the extrapolation volume. But, in order to take previous and future frames into account, the samples are taken from the upscaled sequence $v_u[x, y, t]$, shifted according to the estimated motion. In doing so, the areas from the previous and future frames can be positioned with fractional-pel accuracy and therewith the real motion of an object can be taken into account more precisely. With that, signal $f[m, n, p]$ in extrapolation volume $L_a$ is given by:

$$f[m, n, p] = v[x_0 - B + m, y_0 - B + n, \tau], \text{ for } p = N_p$$

14
and for $p = 0, \ldots, (N_p + N_f) \backslash N_p$:

$$f[m, n, p] = v_{u}[D(x_0 - B) + \hat{x}_d^{(p-N_p)} + Dm, D(y_0 - B) + \hat{y}_d^{(p-N_p)} + Dn, \tau - N_p + p]. \quad (28)$$

In the next step, the model of the signal in the aligned volume $\mathcal{L}_a$ is generated by means of 3D-FSE-OD in order to extrapolate the signal from the correctly received areas into the lost area. Since the motion is revoked by aligning the extrapolation volume prior to the model generation the shortcoming of 3D-FSE, respectively 3D-FSE-OD is resolved and an improved extrapolation quality can be achieved. Finally, the samples of the model corresponding to loss area $B$ are cut out of the model and are used for concealing the distortion.

4. Results

4.1. Simulation Setup

For testing the abilities of MC-FSE, we implemented the algorithm into the H.264/AVC decoder of reference software JM14.0 [38]. An overview of the H.264/AVC codec can be found in [39, 40, 41]. In addition to MC-FSE, Temporal Replacement (TR) [11], EBMA [12], DMVE [13], MH-EC [18], IFS [16], the original 3D-FSE [23] and the enhanced 3D-FSE-OD are implemented in order to compare the performance with other error concealment algorithms. None of the regarded algorithms makes use of an explicit scene change detection. In general, all temporal and spatio-temporal error concealment algorithms suffer from scene changes, as in such a case the reference frames can contain different content compared to the distorted frame. But as we focus on examining the extrapolation quality in our experiments and want to avoid influences from a scene change detection algorithm and intra-frame concealment, which becomes necessary if scene changes occur, we do not include a scene change detection in our experiments. In addition to that, the examined sequences only contain few scene changes and this handicap is present for all the regarded algorithms in the same way.
For simulating the losses and obtaining the results listed in the subsequent tables, three different error patterns are imprinted on the just decoded frames at the decoder. These losses are concealed by the various error concealment algorithms. If no error concealment is applied, the distorted and uncorrected frame is copied to the buffer of decoded frames, causing severe error propagation. Fig. 10 b) shows an example image, suffering from the imprinted error pattern as well as error propagation from previous frames. We prefer this artificial method over using a channel simulator on the coded bitstream since inserting errors into a coded bitstream produces unpredictable errors in the decoded frames. Changing a bit in the data stream of one sequence might cause that a whole frame cannot be decoded at all whereas in another sequence a flipped bit at the same position only corrupts one single block. Since we aim at evaluating the actual signal extrapolation performance within the scope of our research, having the same error patterns for the different examined sequences is indispensable. Nevertheless, we choose the error patterns to be close to typical errors occurring if an H.264/AVC coded bitstream gets distorted. As shown in [2], H.264/AVC offers the possibility to divide a frame into several slices that can be encoded and decoded independently. The standard further contains Flexible Macroblock Ordering (FMO) which controls the assignment of macroblocks to the different slices. Fig. 6 shows two typical FMO patterns consisting of four slice groups. If the bits of one of the slice groups gets distorted by a transmission error, the macroblocks contained in this slice group become corrupted. Considering the FMO patterns from Fig. 6, a transmission error results either in isolated block losses or consecutive block losses. Hence, the first error pattern that we consider is called DISPERSED (see Fig. 7 a) ) and consists of isolated block losses of size $16 \times 16$ samples, occurring with a distance of 16 samples between them. The second error pattern is called INTERLEAVED (see Fig. 7 b) ) and is used for simulating consecutive block losses where several lines of macroblocks get lost within one frame, either located at the left side of the frame or shifted to the right. The third error pattern is called MIXED and consists of isolated as well as consecutive block losses. In every other frame, the error patterns are shifted in vertical and horizontal direction. For the evaluation, the CIF-sequences “Vimto”, “Hockey”, “Basketball”, “Canoe”, “Fast Food” and “Foreman” are regarded. The sequences are coded with QP28 in IPPP order with a group of pictures (GOP) length of 12 frames for concealing errors in P-frames and in IPBP order with a GOP length of 23 frames for concealing errors in B-frames. The sequences have a length of 216 frames or 184 frames, respectively. Thereby, the errors are imprinted either in the P-frames or in the B-frames only. Regarding errors in B-frames, the special case arises, that no error propagation can occur and past and future frames can be used for concealing the losses. For evaluating the quality of the concealment, the Peak Signal to Noise Ratio (PSNR) of the luminance component is used as measure. In this context, PSNR is evaluated against the coded sequence and only over regions that are suffering either from the imprinted errors or from error propagation. In doing so, the extrapolation quality can be evaluated more accurately since the undistorted samples are not taken into account. Although PSNR is not a good metric to
evaluate the absolute quality of a sequence, it is suited well for comparing the relative quality. According to [42], PSNR is highly correlated to visual quality if similar distortions are regarded within one sequence. So, if for one sequence an algorithm can achieve a higher PSNR compared to another one, this also corresponds to an improved visual quality of the concealment.

The motion estimation for MC-FSE is performed with full-pel, half-pel and quarter-pel accuracy. If half-pel and quarter-pel accuracy is used, the twice and fourfold upscaled reference frames are generated by the same procedure as used in H.264/AVC. There, the half-pel positions are calculated by using a 6-tap filter in horizontal and vertical direction. The quarter-pel positions are calculated by bilinear interpolation between the two nearest full- and half-pel samples. DMVE and EBMA use a full search for the candidate blocks and the search is carried out also with full-pel, half-pel, and quarter-pel accuracy in the same way as for MC-FSE in order to achieve a fair comparison.

4.2. Parameter Selection

For carrying out the simulations the following parameter set is used. Most of the parameters that are necessary for the model generation for 3D-FSE, 3D-FSE-OD and MC-FSE are selected according to the originally proposed 3D-FSE in [23]. Hence, the used basis functions are the functions of the three-dimensional discrete Fourier transform. As has been shown in detail in [8] and [23], this set of basis functions is suited well for extrapolating all kinds of content, smooth as well as noise like areas and edges. In addition to that, an efficient implementation in the transform domain can be used for these functions. For this, an FFT of size $64 \times 64 \times 16$ is used for the transform into the Fourier domain. The support volume $A$ has a width of $B = 16$ samples, and as far as available $N_p = 2$ previous frames are used for reference when concealing P-frames and, if possible, $N_p = 2$ previous and $N_f = 1$ following frames are used for reference when concealing B-frames. The decay factor of the weighting function is set to $\hat{\rho} = 0.8$, and if consecutive block losses are concealed, the already concealed blocks are weighted by the additional factor $\delta = 0.2$. Whereas only 200 iterations are carried out for 3D-FSE for generating the model, the
number of iterations for 3D-FSE-OD and MC-FSE is set to 800. The increased number of iterations is necessary as 3D-FSE-OD and MC-FSE make use of orthogonality deficiency compensation which requires more iterations for generating the model. In this context, the orthogonality deficiency compensation factor $\gamma$ is set to 0.7. Selecting these values for $\gamma$ and the number of iterations, a good trade-off between extrapolation quality and computational complexity can be met. As shown in [30], the extrapolation quality could be further improved slightly by using smaller values for $\gamma$ and increasing the number of iterations, but we abstained from this option in order to keep the computational load manageable.

For all algorithms that make use of motion estimation, a full search is performed with a maximum search range of $d_{\text{max}} = 16$ samples. The decision area $\mathcal{M}$ used for MC-FSE is a border of 4 samples width. As shown in [13], a small width results in a low complexity but also a reduced estimation accuracy. In contrast to this, a large width increases the estimation accuracy but also increases the computational load. Thus, we use a border width of 4 samples as a trade-off between estimation quality and computational complexity.

The two thresholds $T_{\text{rel}}$ and $T_{\text{abs}}$ which MC-FSE uses for evaluating the motion estimation are set to $T_{\text{rel}} = 3$ and $T_{\text{abs}} = 10$. For deriving these parameters, we have tested a lot of different parameter sets on the sequences “Discovery City” and “Discovery Orient”. These sequences are suited well, as they contain very different scenes, some with fast and some with slow motion, as well as some with homogenous and others with textured content. Fortunately, the exact values of the listed parameters are not very critical. That is to say, the parameters can be varied widely without affecting the extrapolation quality heavily and none of the parameters depends on certain sequences. To give an example, if the parameters $T_{\text{rel}}$ and $T_{\text{abs}}$ are varied between $1.5 < T_{\text{rel}} < 4.5$ and $5 < T_{\text{abs}} < 20$ respectively, the extrapolation quality for error pattern DISPERSED in P-frames for these two sequences only varies between $-0.06$ dB and $+0.19$ dB. But if parameters far away from the above listed ones are regarded, the extrapolation quality drops significantly. If very large values are considered, the motion vectors are selected every time and a misalignment of the extrapolation cannot be prevented, resulting in a reduced extrapolation quality. But if very small values are considered, the motion estimation is always regarded as unreliable and the performance of MC-FSE drops to the level of 3D-FSE-OD. Hence, selecting the parameters from the above given ranges provides a good trade-off between the risk of misaligning the extrapolation volume and effectively revoking the estimated motion for setting up a more suited extrapolation volume.

4.3. Objective Quality Evaluation

In Table 1, the obtainable PSNR values are listed for concealing error pattern DISPERSED in P-frames by using the above listed error concealment techniques. In order to concentrate the results and to show the other scenarios as well, Table 2 lists the gain in PSNR each error concealment technique can achieve against DMVE, averaged over all sequences. Thereby, the error patterns
Table 1: Extrapolation quality for 216 Frames, GOP size 12, P-Frames, QP28 and error pattern DISPERSED

| Winter | Hockey | Basketball | Burger | Fast Food | Foreman |
|--------|--------|------------|--------|-----------|---------|
| TH [12]| 21.59 dB| 17.78 dB | 16.74 dB| 19.97 dB  | 17.35 dB|
| DMVE  [13]| 22.78 dB| 23.51 dB | 20.97 dB| 22.69 dB  | 23.02 dB|
| DMVE HPQ| 25.54 dB| 24.74 dB | 20.42 dB| 22.89 dB  | 24.24 dB|
| EBMA [12]| 27.38 dB| 23.30 dB | 19.11 dB| 22.48 dB  | 24.51 dB|
| EBMA HPQ| 27.39 dB| 24.37 dB | 19.37 dB| 22.59 dB  | 23.57 dB|
| MC-EC [14]| 27.09 dB| 24.33 dB | 20.84 dB| 23.59 dB  | 24.06 dB|
| TFS [15]| 29.28 dB| 24.41 dB | 20.92 dB| 24.31 dB  | 24.69 dB|
| DMVE QPel| 30.93 dB| 25.52 dB | 20.89 dB| 23.74 dB  | 26.11 dB|
| DMVE HPQ| 29.09 dB| 29.30 dB | 21.66 dB| 25.91 dB  | 26.80 dB|
| MC-EC HPQ| 30.42 dB| 27.22 dB | 21.69 dB| 24.79 dB  | 26.96 dB|
| MC-EC QPel| 30.58 dB| 27.69 dB | 21.89 dB| 24.82 dB  | 26.69 dB|

Table 2: Mean gain vs. DMVE for different error scenarios

| Table | 3Pel | Quarter-Pel | Half-Pel | 1 Pel | 0.5 Pel | 0 Pel | 0.25 Pel | 0.125 Pel | 0 Pel | 0.125 Pel |
|-------|------|-------------|---------|-------|---------|------|---------|-----------|------|----------|
| Table 2 | 5 Pel | Quarter-Pel | Half-Pel | 1 Pel | 0.5 Pel | 0 Pel | 0.25 Pel | 0.125 Pel | 0 Pel | 0.125 Pel |

Table 3: Mean gain produced by fractional-pel motion estimation

| DMVE | EBMA | MC-FSE |
|------|------|-------|
| P-Concealment | DISPERSED | Gain Half-Pel | Gain Quarter-Pel | Gain Half-Pel | Gain Quarter-Pel |
| DISPERSED | 24.49 dB | 24.88 dB | 27.21 dB | 0.83 dB | 0.41 dB |
| INTERLEAVED | 20.55 dB | 19.84 dB | 25.21 dB | 0.41 dB | 0.41 dB |
| MIXED | 22.58 dB | 21.27 dB | 24.55 dB | 0.41 dB | 0.41 dB |
| B-Concealment | DISPERSED | Gain Half-Pel | Gain Quarter-Pel | Gain Half-Pel | Gain Quarter-Pel |
| DISPERSED | 24.94 dB | 25.96 dB | 28.84 dB | 0.59 dB | 0.59 dB |
| INTERLEAVED | 22.23 dB | 21.58 dB | 26.59 dB | 0.59 dB | 0.59 dB |
| MIXED | 24.49 dB | 22.25 dB | 27.22 dB | 0.59 dB | 0.59 dB |

19
DISPERSED, INTERLEAVED and MIXED are applied on P- and B-frames. Regarding the two tables, it becomes obvious that TR is inferior to all compared algorithms, but it is also by far the most simple one. The two temporal algorithms EBMA and DMVE provide a decent extrapolation quality, while DMVE is superior to EBMA in most cases. This is especially true when regarding error patterns INTERLEAVED and MIXED. In this case, EBMA cannot estimate the motion well and falls behind DMVE. But, both algorithms can also significantly benefit from motion compensation with fractional-pel accuracy. In Table 3 the gain achievable by fractional-pel motion estimation is listed for the different scenarios. So, for DMVE a gain of up to 0.58 dB can be achieved if motion estimation is performed with quarter-pel accuracy, for EBMA a gain of up to 0.41 dB is possible.

Regarding the spatio-temporal algorithms MH-EC, IFS and 3D-FSE, which are used for reference, one can see that taking the spatial neighborhood into account can significantly improve the extrapolation quality. In this context, 3D-FSE exploits the additional information more effectively than the two other algorithms and outperforms them by 0.50 to 2.26 dB PSNR in the mean. Taking a look at 3D-FSE-OD, it becomes also obvious that the incorporation of orthogonality deficiency compensation into 3D-FSE leads to a further improvement and 3D-FSE-OD is able to outperform the original 3D-FSE by 0.76 to 0.94 dB.

When evaluating the gain that can be achieved by estimating the motion and aligning the extrapolation volume, one has to compare 3D-FSE-OD and MC-FSE. In doing so, one can recognize that for P-frames approximately the same performance can be obtained if motion estimation and volume alignment is performed only with full-pel accuracy. This is due to the fact that although the extrapolation quality is improved for most blocks by aligning the volume prior to the model generation, in some cases the volume can get slightly misaligned, despite of the motion estimation evaluation. In such a case, the extrapolation quality of MC-FSE can be inferior to 3D-FSE-OD and the improvement gained so far is consumed. But, by using quarter-pel accuracy the extrapolation quality can be improved in the mean by at least 0.39 dB. Hence, MC-FSE is superior to all other considered error concealment algorithms if half-pel or quarter-pel accuracy is used for motion estimation and volume alignment. The improvement further increases if B-frames are considered. For B-frames, MC-FSE outperforms 3D-FSE-OD by up to 0.74 dB if quarter-pel accuracy is applied. The gain over DMVE grows to up to 4.49 dB. The very high extrapolation quality of 3D-FSE, 3D-FSE-OD and MC-FSE for B-frames can be explained by the fact that there previous and future frames are available. Due to this, the lost blocks are surrounded by correctly received blocks in spatial and temporal direction which is very helpful for the model generation. Comparing MC-FSE to the original 3D-FSE, a gain of up to 1.64 dB can be noticed, which arises in B-frames for error pattern MIXED. But, similar gains can be achieved for the other error patterns as well. If the results of 3D-FSE-OD are also taken into account, one can recognize that approximately half the gain is produced by applying orthogonality deficiency compensation to the three-dimensional model generation and the other half is produced by motion estimation and volume alignment.
with fractional-pel accuracy. In addition to the above mentioned DMVE and EBMA, Table\textsuperscript{3} lists the gain in extrapolation quality that can be achieved for MC-FSE by using motion estimation with fractional-pel accuracy. Apparently, the gain achievable by fractional-pel motion estimation is similar for MC-FSE as for the aforementioned algorithms and by using half-pel motion estimation, the extrapolation quality can be improved by up to 0.53 dB, by using quarter-pel accuracy by up to 0.64 dB.

Fig.\textsuperscript{8} shows the percentage of discarded motion vectors per frame for sequence “Vimto” with error pattern DISPERSED in P-frames if MC-FSE is carried out with full-pel, half-pel, and quarter-pel accuracy. Apparently, in some frames approximately all the motion vectors are used for aligning the extrapolation volume whereas for other frames nearly all motion vectors are discarded and the fixed extrapolation volume is regarded. The first 93 frames of sequence “Vimto” consist of two scenes with predominant translational motion. During these scenes the motion can be estimated well and be revoked for improving the model generation. At the transition between the two scenes at frame 54, no reliable motion can be found and MC-FSE falls back to 3D-FSE-OD for almost all blocks, leading to a similar quality for this frame. The second half of the sequence contains a lot of non-translational motion. There, a successful alignment is only possible for few blocks and the motion estimation is dropped in favor of a fixed extrapolation volume. Taking this figure into account it can be discovered that the motion estimation reliability check in MC-FSE behaves as desired and that the alignment of the volume only takes place if it is reasonable.

Although the simulation results presented so far are for sequences coded with
QP28, our experiments have shown that the insights are still valid for stronger and weaker compression levels. To illustrate this, Table 4 shows the results for sequence “Vimto” coded with different QPs. Here, again error pattern DISPERSED is concealed in P-frames. Since the PSNR is evaluated against the undistorted coded sequence, the extrapolation quality slightly increases towards larger QPs, independently of the concealment algorithm. This is due to the fact that the amount of detail per block decreases due to quantization. The parameters for MC-FSE and 3D-FSE-OD are the same as mentioned above and are not adapted to the individual QPs as our simulations have shown that an adaption is not required.

In order to quantify the computational overhead produced by motion estimation and volume alignment and to compare the different algorithms, Table 5 lists the extrapolation time per block for the regarded algorithms. The simulations are carried out on one core of an Intel Core2 Quad @ 2.83 GHz. The complete extrapolation process is performed in MATLAB R2007b. To achieve this, the data necessary for concealing a block first is transferred from the JM reference decoder to MATLAB and after extrapolation, the concealed block is retrans-
ferred to the reference decoder. Obviously, the algorithms based on Frequency Selective Extrapolation possess an extrapolation time that is significantly higher than the other algorithms. But, as shown above, at the same time, they also provide a significantly improved extrapolation quality. Comparing 3D-FSE-OD and MC-FSE with full-pel accuracy, the amount of time to conceal a block is similar. But, if a higher accuracy is used for motion estimation, the number of candidate positions to examine for estimating the motion increases quadratically. For our setup, using quarter-pel accuracy in summary increases the extrapolation time by 32%. But, regarding the improved extrapolation quality, this increment should be tolerable.

4.4. Visual Quality Evaluation

In addition to the extrapolation quality of the distorted regions compared to the coded sequence, the quality of the concealed sequence compared to the uncoded sequence is of interest as well, since the consideration of whole frames expresses the viewers impression of a sequence more. To account for this, Fig. 9 shows the PSNR per frame against the uncoded sequence for concealment by EBMA with quarter-pel accuracy, DMVE with quarter-pel accuracy, MH-EC, IFS, 3D-FSE, 3D-FSE-OD, and MC-FSE with quarter-pel accuracy. Here, sequence “Vimeo” coded with QP28 is regarded and error pattern DISPERSED is concealed in P-frames. As there are only errors in P-frames considered, the quality of the concealed sequence is high at the beginning of each GOP and then decreases within the GOP due to error propagation. Furthermore, the scene changes at frame 54 and 94 cause a drop of the extrapolation quality, independent of the actual error concealment algorithm. The decay is stronger for temporal error concealment than for spatio-temporal. While spatio-temporal algorithms can also exploit information from the current frame for concealment, temporal ones have to completely rely on previous frames that may be totally distinct in the case of a scene change. Regarding the model generation based algorithms, the extrapolation quality of 3D-FSE-OD and MC-FSE is similar for many frames, but there are also a lot of frames where MC-FSE is superior. This happens mainly in frames with high motion. There, the gain of MC-FSE over 3D-FSE and 3D-FSE-OD often grows to more than 2 dB.

For error concealment, besides the objective measure the visual extrapolation quality is important. In order to illustrate the visual quality, Fig. 10 shows frame 90 of test sequence “Fast Food” with different concealment techniques applied to. Subfigure a) shows the undistorted frame, whereas b) shows the frame for which no error concealment is applied. Hence, the frame is suffering from the imprinted error pattern and from error propagation from previous frames. In subfigures c) to e) TR, EBMA and DMVE are used for error concealment. Here, motion compensation with quarter-pel accuracy is used for the latter two. Subfigures f) to h) result from application of IFS, MH-EC and the original 3D-FSE as concealment algorithms. The last two subfigures i) and j) result from error concealment by the proposed 3D-FSE-OD and MC-FSE with quarter-pel motion estimation. Obviously, the frames concealed by TR, EBMA, DMVE are heavily suffering from block artifacts. This is due to the fact that the woman’s
Figure 9: PSNR per frame for sequence “Vimto” distorted by error pattern DISPERSED. The PSNR is evaluated for whole frames and against the uncoded original sequence.

arm is moving very fast and that EBMA and DMVE cannot find a fitting block in the previous frame. But, the frames concealed by IFS and MH-EC are also suffering from block artifacts and the losses cannot be concealed well. Regarding 3D-FSE, 3D-FSE-OD and MC-FSE which utilize the three-dimensional model generation, no block artifacts can be found. The visual extrapolation quality of these three algorithms is very high and is superior to the other considered algorithms. However, in direct comparison, the frame resulting from MC-FSE shows less artifacts and more details compared to the ones resulting from 3D-FSE-OD. 3D-FSE-OD itself provides a superior visual quality compared to 3D-FSE which does not make use of orthogonality deficiency compensation. Altogether, the visual results are reflective for the above outlined PSNR measurements.

5. Conclusion

In this contribution, Motion Compensated Three-Dimensional Frequency Selective Extrapolation was presented for error concealment in video communication. This algorithm utilizes Three-Dimensional Frequency Selective Extrapolation which is further enhanced by orthogonality deficiency compensation for generating a model of the signal suffering from distortion. In doing so, the signal is extrapolated from correctly received regions into the distorted regions. In advance of the model generation, the motion in a video sequence is estimated, evaluated, and compensated with fractional-pel accuracy. The introduced algorithm provides a very high objective as well as subjective extrapolation quality and is able to outperform the original Three-Dimensional Frequency Selective Extrapolation error concealment algorithm significantly. The proposed exten-
Figure 10: Visual error concealment results for frame 90 of test sequence “Fast Food”. a) Undistorted compressed frame, b) Frame suffering from distortion and error propagation. Error concealment by c) TR, d) EBMA QPel, e) DMVE QPel, f) IFS, g) MH-EC, h) 3D-FSE, i) 3D-FSE-OD, j) MC-FSE QPel.
sion of orthogonality deficiency compensation to the three-dimensional model generation and especially incorporating motion compensation into the model generation yields a gain of up to 1.64 dB PSNR. Regarding the motion estimation accuracy, a gain of up to 0.64 dB can be achieved by performing motion estimation with fractional-pel accuracy instead of full-pel accuracy.

We also showed that typical error concealment algorithms like the Decoder Motion Vector Estimation or the Extended Boundary Matching Algorithm can be improved by up to 0.58 dB PSNR by using motion estimation with fractional-pel accuracy if the algorithms are carried out with a full search.

Besides error concealment in video sequences, the proposed algorithm could be used as well for concealing errors in multi-view sequences in the future. There, in addition to motion, the disparity between different views should compensated as well if the Four-Dimensional Frequency Selective Extrapolation from [43] is used. Since disparity as well as the here regarded motion does not necessarily have to follow the sampling grid, the alignment of the extrapolation volume in temporal and view direction can also also be improved by using fractional-pel accuracy. A more precisely aligned four-dimensional hyper-volume can serve as better basis for the model generation and therewith increase the concealment quality.

References

[1] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264 — ISO/IEC 14496-10 AVC), JVT-G050r1, 2003.

[2] T. Stockhammer, M. M. Hannuksela, H.264/AVC video for wireless transmission, IEEE Wireless Communications 12 (2005) 6–13.

[3] Y. Wang, Q.-F. Zhu, Error control and concealment for video communication: a review, Proceedings of the IEEE 86 (1998) 974–977.

[4] Y. Wang, Q.-F. Zhu, L. Shaw, Maximally smooth image recovery in transform coding, IEEE Trans. on Commun. 41 (1993) 1544–1551.

[5] H. Sun, W. Kwok, Concealment of damaged block transform coded images using projections onto convex sets, IEEE Trans. Image Process. 4 (1995) 470–477.

[6] Z. Alkachouh, M. Bellanger, Fast DCT-based spatial domain interpolation of blocks in images, IEEE Trans. Image Process. 9 (2000) 729–732.

[7] X. Li, M. T. Orchard, Novel sequential error-concealment techniques using orientation adaptive interpolation, IEEE Trans. Circuits Syst. Video Technol. 12 (2002) 857–864.

[8] A. Kaup, K. Meisinger, T. Aach, Frequency selective signal extrapolation with applications to error concealment in image communication, Int. J. Electron. Commun. (AEÜ) 59 (2005) 147–156.
[9] S. Tsekeridou, I. Pitas, MPEG-2 error concealment based on block-matching principles, IEEE Trans. Circuits Syst. Video Technol. 10 (2000) 646–658.

[10] W.-Y. Kung, C.-S. Kim, C.-C. J. Kuo, Spatial and temporal error concealment techniques for video transmission over noisy channels, IEEE Trans. Circuits Syst. Video Technol. 16 (2006) 789–803.

[11] A. S. Bopardikar, O. I. Hillestad, A. Perkis, Temporal concealment of packet-loss related distortions in video based on structural alignment, in: Proc. of the Eurescom summit 2005, Heidelberg, Germany, 2005.

[12] W.-M. Lam, A. R. Reibman, B. Liu, Recovery of lost or erroneously received motion vectors, in: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Minneapolis, USA, 1993, pp. 417–420.

[13] J. Zhang, J. F. Arnold, M. F. Frater, A cell-loss concealment technique for MPEG-2 coded video, IEEE Trans. Circuits Syst. Video Technol. 10 (2000) 659–665.

[14] M. Friebe, A. Kaup, Spatio-temporal fading scheme for error concealment in block-based video decoding systems, in: Proc. Int. Conf. on Image Processing (ICIP), 2006, pp. 2237–2240.

[15] M. Friebe, A. Kaup, Spatio-bi-temporal error concealment in block-based video decoding systems, in: Proc. Int. Workshop on Multimedia Signal Processing, 2006, pp. 296–301.

[16] M.-C. Hwang, J.-H. Kim, C.-S. Park, S.-J. Ko, Improved fading scheme for spatio-temporal error concealment in video transmission, IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences E91-A (2008) 740–748.

[17] Y. Chen, O. Au, C. Ho, J. Zhou, Spatio-temporal boundary matching algorithm for temporal error concealment, in: IEEE Int. Symposium on Circuits and Systems, 2006.

[18] K. Song, T. Chung, C.-S. Kim, Y.-O. Park, Y. Kim, Y. Joo, Y. Oh, Efficient multi-hypothesis error concealment technique for H.264, in: IEEE Int. Symposium on Circuits and Systems, 2007, pp. 973–976.

[19] M. Flierl, B. Girod, Multihypothesis motion estimation for video coding, in: Proc. Data Compression Conference, 2001, pp. 341–350.

[20] S. Belfiore, M. Grangetto, E. Magli, G. Olmo, Spatiotemporal error concealment with optimized mode selection and application to H.264, Signal Processing: Image Communication 18 (2003) 907–923.

[21] O. Hadar, M. Huber, R. Huber, S. Greenberg, New hybrid error concealment for digital compressed video, EURASIP Journal on Applied Signal Processing 1 (2005) 1821 – 1833.
[22] L. Atzori, F. G. B. De Natale, C. Perra, A spatio-temporal concealment technique using boundary matching algorithm and mesh-based warping (BMA-MBW), IEEE Trans. on Multimedia 3 (2001) 326–338.

[23] K. Meisinger, A. Kaup, Spatiotemporal selective extrapolation for 3-D signals and its applications in video communications, IEEE Trans. Image Process. 16 (2007) 2348–2360.

[24] B. A. Olshausen, D. J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1, Vision Research 37 (1997) 3311–3325.

[25] J. A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Information Theory 50 (2004) 2231–2242.

[26] V. Temlyakov, Weak greedy algorithms, Advances in Computational Mathematics 12 (2000) 213–227.

[27] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (1993) 3397–3414.

[28] B. K. Natarajan, Sparse approximate solutions to linear systems, SIAM Journal on Computing 24 (1995) 227–234.

[29] J. Seiler, K. Meisinger, A. Kaup, Orthogonality deficiency compensation for improved frequency selective image extrapolation, in: Proc. Picture Coding Symposium (PCS), Lisboa, Portugal, 2007.

[30] J. Seiler, A. Kaup, Fast orthogonality deficiency compensation for improved frequency selective image extrapolation, in: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, USA, 2008, pp. 781–784.

[31] J. Seiler, A. Kaup, Motion compensated frequency selective extrapolation for error concealment in video coding, in: Proc. European Signal Processing Conference, Lausanne, Switzerland, 2008.

[32] S. Ericsson, Fixed and adaptive predictors for hybrid predictive/transform coding, IEEE Trans. on Commun. 33 (1985) 1291–1302.

[33] B. Girod, The efficiency of motion-compensating prediction for hybrid coding of video sequences, IEEE Journal Selected Areas in Communications 5 (1987) 1140–1154.

[34] B. Girod, Motion-compensating prediction with fractional-pel accuracy, IEEE Trans. on Commun. 41 (1993) 604–612.

[35] M. Funagalli, R. Lancini, S. Tubaro, A sequence-based error-concealment algorithm for an unbalanced multiple description video coding system, Image Communication 21 (2006) 829–849.
[36] C. S. Park, J. Ye, S. U. Lee, Lost motion vector recovery algorithm, in: IEEE Int. Symposium on Circuits and Systems, volume 3, 1994, pp. 229–232.

[37] T. Wedi, Adaptive interpolation filters and high-resolution displacements for video coding, IEEE Trans. Circuits Syst. Video Technol. 16 (2006) 484–491.

[38] Joint Video Team, H.264/AVC reference software (JM 14.0), http://iphome.hhi.de/suehring/tml/, 2008.

[39] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 560–576.

[40] G. J. Sullivan, T. Wiegand, Video compression - from concepts to the H.264/AVC standard, Proceedings of the IEEE 93 (2005) 18–31.

[41] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Trans. Circuits Syst. Video Technol. 4 (2004) 7–28.

[42] T. Oelbaum, K. Diepold, W. Zia, A generic method to increase the prediction accuracy of visual quality metrics, in: Proc. Picture Coding Symposium (PCS), Lisboa, Portugal, 2007.

[43] U. Fecker, J. Seiler, A. Kaup, 4-D frequency selective extrapolation for error concealment in multi-view video, in: Proc. Int. Workshop on Multimedia Signal Processing, Cairns, Australia, 2008, pp. 267–272.