Motion-Compensated Coding and Frame Rate Up-Conversion: Models and Analysis

Yehuda Dar and Alfred M. Bruckstein

***Abstract*— Block-based motion estimation (ME) and motion compensation (MC) techniques are widely used in modern video processing algorithms and compression systems. The great variety of video applications and devices results in diverse compression specifications, such as frame rates and bit rates. In this paper, we study the effect of frame rate and compression bit rate on block-based ME and MC as commonly utilized in inter-frame coding and frame rate up-conversion (FRUC). This joint examination yields a theoretical foundation for comparing MC procedures in coding and FRUC. First, the video signal is locally modeled as a noisy translational motion of an image. Then, we theoretically model the motion-compensated prediction of available and absent frames as in coding and FRUC applications, respectively. The theoretic MC-prediction error is studied further and its autocorrelation function is calculated, yielding useful separable-simplifications for the coding application. We argue that a linear relation exists between the variance of the MC-prediction error and temporal distance. While the relevant distance in MC coding is between the predicted and reference frames, MC-FRUC is affected by the distance between the frames available for interpolation. We compare our estimates with experimental results and show that the theory explains qualitatively the empirical behavior. Then, we use the models proposed to analyze a system for improving of video coding at low bit rates, using a spatio-temporal scaling. Although this concept is practically employed in various forms, so far it lacked a theoretical justification. We here harness the proposed MC models and present a comprehensive analysis of the system, to qualitatively predict the experimental results.**

***Index Terms*— Frame rate up-conversion, motion compensation, motion compensated interpolation, video coding.**

I. Introduction

TEMPORAL redundancy is a main property of video signals. This redundancy originates in similarity between successive frames in a video scene. A video scene can be thought of as a composition of static and moving regions. Therefore, many video compression and processing systems utilize motion estimation (ME). Ideally, the motion should be estimated per pixel; however, practical systems have severe resource and run-time limitations, and therefore cannot apply estimation per pixel. Hence, block-based ME techniques are widely used in practical video compression and processing algorithms. Block-based ME is the procedure of estimating block motion by comparing it with blocks in a search area within another frame in the sequence. This method approximates the motion as translational, and represents it by a motion-vector and a reference frame indication.

In block-based hybrid compression, ME is utilized for inter-frame prediction of a coded block. Motion-compensation (MC) uses the difference between the coded block and its prediction. This results in block prediction errors, also known as MC-residual. The MC-residual is coded further and sent to the decoder. Therefore, the MC-residual greatly affects the performance of inter-frame coding. Furthermore, due to the extensive use of inter-frame coding, the MC-residual significantly influences the overall compression performance.

Accordingly, the MC-residual has been widely studied since the 1980's [1]–[12]. However, these studies have not explicitly considered the frame-rate effect on the MC-residual statistics. Moreover, only Guo et al. [10] mentioned the influence of frame-reconstruction quality (and, therefore, bit-rate) on the MC-residual. In this paper, we analyze the effects of frame-rate (through the temporal-distance) and bit-rate on the MC-residual autocorrelation function. The available models are too complex for use as a basis for analysis of an entire compression system. We here propose two models for deriving the autocorrelation. First, we obtain a rather complex expression from our theoretic model for MC-prediction of an available frame (as done in coding). We then simplify the autocorrelation to a separable form similar to [5] and [11]. We subsequently justify our analysis by experimental observations.

Frame-rate up conversion (FRUC) is the procedure of increasing the frame-rate of a video by temporal interpolation of frames. There are several motivations for using FRUC. It is used for video format conversion when the target format has higher frame-rate. In addition, high frame-rates were found to increase the subjective quality [13]; therefore, some applications apply FRUC on low frame-rate videos. Another application of FRUC is in improving low bit-rate video coding as follows: the frame-rate is reduced before compression, and restored to its original value after the reconstruction of the compressed data. As a result, the output video quality is improved for the given bit-budget.

FRUC algorithms trade off between computational complexity and the quality of the interpolated frames.

Manuscript received April 20, 2014; revised October 29, 2014 and February 11, 2015; accepted February 21, 2015. Date of publication March 11, 2015; date of current version March 31, 2015. This work was supported by the Technion Funds for Security Research. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anthony Vetro.

The authors are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: ydar@cs.technion.ac.il; freddy@cs.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes a pdf that is 1 MB. Contact ydar@tx.technion.ac.il for further questions about this work.

Digital Object Identifier 10.1109/TIP.2015.2412378

Simple FRUC techniques disregard the motion in the sequence, and interpolate by frame repetition or averaging. For non-static regions, this often results in motion jerkiness and ghost artifacts. Therefore, the commonly used interpolation techniques do consider motion. Specifically, methods that utilize motion-trajectory estimation are known as motion-compensated FRUC (MC-FRUC).

Some studies have proposed complex FRUC algorithms that try to accurately model the motion in the video, see [14]. However, high computational complexity limits these algorithms for offline usage, whereas some applications require real-time FRUC. A reasonable computational complexity is achieved in block-based MC-FRUC techniques; therefore, they are widely used and studied [15]–[19]. Block-based MC-FRUC is usually performed by applying block-matching procedure between existing frames, resulting in a trajectory of the estimated translational motion; then, this motion-trajectory is used for interpolating missing blocks according to the applied method [15]–[19].

In [19], the MC-FRUC error was analyzed in the power-spectral-density (PSD) domain and by using a statistical model of the motion-vector error. Dane and Nguyen then searched for the optimal temporal filter. In this paper, we study the block-based MC-FRUC error in the pixel domain. The examined procedure models low-complexity methods (see [15]), which are commonly used. Consequently, the proposed analytic derivations are relatively simple.

Block-based ME differs from the true motion by assuming it is translational. This sub-optimality has minor importance in the application of MC for inter-frame coding, where the motion estimation is performed at the encoder between two accessible frames, and the target is minimal prediction residual. However, ME in FRUC aims at estimating the true motion in a missing frame. Therefore, the translational motion assumption deteriorates MC-FRUC performance. Dane and Nguyen [19] discuss the differences between the application of MC to coding and FRUC. This paper continues this examination by providing a side by side analysis of MC-coding and MC-FRUC, which are readily comparable due to joint assumptions and mathematical tools.

In the last part of this work, we use the proposed models in adapting a previous analysis for image compression at low bit-rates [20] to video signals. The approach suggests to improve compression using spatio-temporal down-scaling before compression and a corresponding up-scaling afterwards, while the codec itself is left unmodified. We show, both theoretically and experimentally, that at low bit-rates, we benefit from applying spatio-temporal scaling. The analysis presented relies on the models proposed in the first part of this paper; specifically, the models for MC-prediction of available and absent frames are used to analyze the low bit-rate compression and temporal up-scaling, respectively. In this paper, for the first time in the literature on video coding, we qualitatively predict the typical performance trade-off curves obtained in practice, based on some very reasonable assumptions on video signal behavior.

This paper is organized as follows. In section II, we present a theoretic model for the video signal. Section III analyzes the MC-prediction and its error for the cases of available and absent frames, i.e., coding and FRUC, respectively. In section IV we study the theoretic estimates using our model. In section V we present experimental results to qualitatively validate our models. Section VI presents a way to use the proposed models in analyzing a system for improvement of video coding at low bit-rates using a spatio-temporal scaling. Section VII concludes this paper.

II. Video Signal Model

A. A Noised Translational Motion Model

The digital video signal is a temporal sequence of 2D images, i.e. $\{f_t(x,y)\}_{t=0}^{T}$. Adjacent frames are known to be correlated; hence, we relate the frames by assuming a translational motion of a 2D image with additive noise process.

We assume that the frame sequence $\{f_t(x,y)\}_{t=0}^{T}$ is decomposable into two sequences. First, a 2D image with a translational motion denoted as $\{v_t(x,y)\}_{t=0}^{T}$. Second, a temporally-accumulated noise process, $\{n_t(x,y)\}_{t=0}^{T}$, that represents differences between $\{v_t(x,y)\}_{t=0}^{T}$ and the actual frames due to deviations from translational motion such as deformations of objects, camera noise or quantization noise. The proposed decomposition is expressed as follows.

$$f_t(x,y) = v_t(x,y) + n_t(x,y) \tag{1}$$

The underlying translational motion process is defined as follows. The motion at the t^{th} frame relative to its predecessor at $t-1$ is denoted as $\varphi(t,t-1) = \left(\varphi_x(t,t-1), \varphi_y(t,t-1)\right)$. Hence, the motion in the video can be represented by the sequence $\{\varphi(i,i-1)\}_{i=1}^{T}$. Moreover, the motion between two time points, t_1 and t_2, is defined as follows.

$$\varphi(t_2,t_1) = \begin{cases} \sum_{i=t_1+1}^{t_2} \varphi(i,i-1), & \text{for } t_1 < t_2 \\ -\sum_{i=t_2+1}^{t_1} \varphi(i,i-1), & \text{for } t_2 < t_1 \\ (0,0), & \text{for } t_1 = t_2 \end{cases} \tag{2}$$

We model v_t to be a constant base frame, v, spatially shifted by $\left(\varphi_x(t,0), \varphi_y(t,0)\right)$, i.e.,

$$v_t(x,y) = v\left(x - \varphi_x(t,0), y - \varphi_y(t,0)\right) \tag{3}$$

The image v is assumed to be a sample of a wide-sense stationary (WSS) process and is modeled using first-order Markov process, its autocorrelation is being given by:

$$R_v(k,l) = \sigma_v^2 \cdot \rho_v^{|k|+|l|}. \tag{4}$$

We model the noise, n_t, as a combination of two elements. First, a temporally-local noise, w_t, is assumed representing distortions that are relevant only for the frame at time t. This component can express various procedures including compression noise (see section II-C), spatial-processing deterioration (e.g., see (52)), or other technical degradations (e.g., camera noise) that can be represented using an appropriate model.

Furthermore, object deformations are represented using a temporally-accumulated noise process. The noise aggregation is assumed to represent a preceding time-interval of

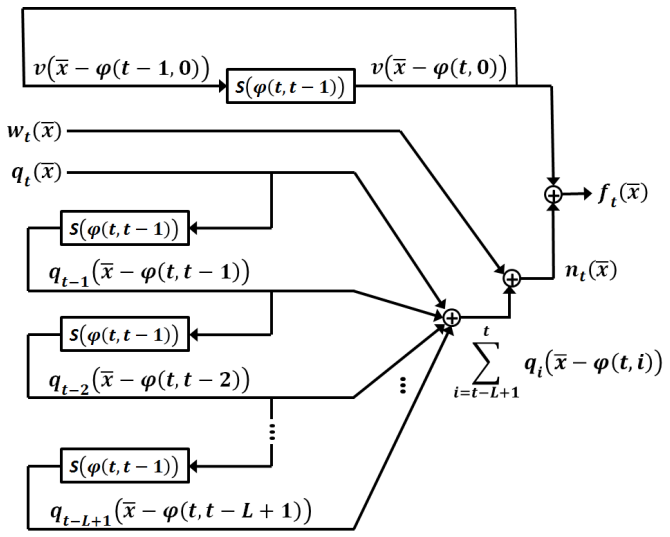


Fig. 1. Demonstration of the proposed video model. The 2D coordinates (x, y) are denoted as $\bar{x}$, and $S(\cdot)$ denotes a 2D-spatial shift.

fixed length. We define the memory parameter of the noise accumulation as the number of temporal samples (i.e., time points) contained in the preceding time-interval, in addition to the current time-point; this amount is denoted by L. The memory parameter depends on temporal properties of the specific video (e.g., motion speed and scene length), and describes them globally in our statistical model. Specifically, the noise represents the accumulated deviation from translational motion during the recent L frames. Our assumptions imply that frames have equal average energy (i.e., f_t has a constant variance for any t). Therefore, there is a fixed amount of object deformation relative to the original form in v; otherwise, the immersion of v in noise will increase over time. For modeling of the n_ts, we use an auxiliary noise sequence $\{q_t\}_{t=-L+1}^{\infty}$, where q_t represents the deviation from translational-motion added at time t (i.e., the disparity in the continuous interval $(t-1, t]$). Moreover, q_t is considered spatially i.i.d, has a zero-mean Gaussian distribution with variance σ_q^2. Furthermore, the noise-sequence's elements are temporally independent, i.e., q_i is independent from q_k for $i \neq k$, and from w_j for any j.

We assume that at each time point, t, the spatial noise signals q_t and w_t affect n_t together with the last $L-1$ preceding q_k elements (Fig. 1), i.e.,

$$n_t(x,y) = w_t(x,y) + \sum_{i=t-L+1}^{t} q_i\left(x-\varphi_x(t,i), y-\varphi_y(t,i)\right). \tag{5}$$

Here we utilized the property $\varphi(t,t) = (0,0)$. Recall that q_i is available also for negative time points starting at $t = -L+1$. Consequently, the temporally-accumulated noise has a spatially i.i.d, zero-mean Gaussian distribution with variance $L \cdot \sigma_q^2$ for any t. Accordingly, $n_t(x,y)$'s autocorrelation is

$$R_{n_t}(k,l) = \left(\sigma_w^2 + L\sigma_q^2\right) \cdot \delta(k,l). \tag{6}$$

Our simplifying assumptions result in a model that represents the noise as a spatially i.i.d process. In contrast, various studies offer more complex models expressing spatial correlation or even non-stationarity [8], [9]. Our framework can accommodate such statistical scenarios by plugging q_i with suitable properties.

Setting (3) and (5) into (1) yields

$$\begin{aligned} f_t(x,y) &= v\left(x-\varphi_x(t,0), y-\varphi_y(t,0)\right) + w_t(x,y) \\ &\quad + \sum_{i=t-L+1}^{t} q_i\left(x-\varphi_x(t,i), y-\varphi_y(t,i)\right). \end{aligned} \tag{7}$$

B. Frame-Rate Effect

The variance of the auxiliary noise elements, σ_q^2, reflects the energy of the differences between successive frames that cannot be perfectly modeled via translational transformations, even for continuous images (i.e., the estimation algorithm has no spatial accuracy issues).

The frame-rate, F_{rate}, defines the time-intervals between successive frames to be $\frac{1}{F_{rate}}$. We assume that the energy of the modifications expressed in σ_q^2 is linear in the temporal-distance. Hence,

$$\sigma_q^2 = \frac{1}{F_{rate}} \cdot \tilde{\sigma}_q^2 \tag{8}$$

where $\tilde{\sigma}_q^2$ is the energy of successive-frames difference for a sequence of one frame per second.

C. Compression Effect

We model quality reduction due to compression as an additional component of the noise w_t. This component is denoted as $w_{t,compression}$ and it is independent of other ingredients of w_t, their sum being denoted as $w_{t,basic}$. As a result,

$$\sigma_w^2 = \sigma_{w,compression}^2 + \sigma_{w,basic}^2 \tag{9}$$

where $\sigma_{w,compression}^2$ is the variance of the compression error, i.e., the mean-squared-error (MSE). $\sigma_{w,basic}^2$ is $w_{t,basic}$'s variance.

We can express $\sigma_{w,compression}^2$ in one of the following ways:

1) Empirical Rate-Distortion Curve:

$$\sigma_{w,compression}^2 = \beta \cdot r^{-\alpha} \tag{10}$$

where, α and β are curve parameters, and r is the bit-rate.

2) Theoretical Rate-Distortion for Memoryless Gaussian Source: A simple theoretical estimate is available under the following assumptions. The compression distortion is similar to the procedure of directly compressing the frame pixels, and the frame pixels constitute a memoryless Gaussian source. The estimate is here given by

$$\sigma_{w,compression}^2 = \sigma_v^2 \cdot 2^{-2r}, \tag{11}$$

where, σ_v^2 is the variance of the Gaussian source, and r is the bit-rate.

3) Given a Value that is Externally Known or Estimated:

$$\sigma^2_{w,compression} = MSE_{compression}. \quad (12)$$

4) Uncompressed Video: An uncompressed video has no compression error, i.e., $\sigma^2_{w,compression} = 0$; hence, $\sigma^2_w = \sigma^2_{w,basic}$.

III. Analysis of Motion-Compensated Prediction

In this section we analyze the two common cases of applying motion-compensation. First, we consider MC-prediction between a pair of available frames, as in MC-coding. Then, we study the case of applying MC-prediction between an existing and absent frames, as in MC-FRUC.

Our analysis is statistical, therefore we differ from practical MC-prediction as follows. First, we consider a single MC procedure in the context of given signal properties, and consider it statistically representative. Second, we assume it is allowed to model signals as functions without explicit spatial boundaries, although obviously a practical MC-prediction has finite extent blocks and search areas.

A. MC-Prediction of an Available Frame

Let us consider the MC-prediction of frame f_t using f_{t-i} as a reference frame, where $i \in \{0, ..., t-1\}$. The prediction relies on estimating the motion between $t-i$ and t using the corresponding frames. This estimate assumes translational motion and is denoted as $\hat{\varphi}(t, t-i \,|\, f_t, f_{t-i}) = \left(\hat{\varphi}_x(t, t-i \,|\, f_t, f_{t-i}), \hat{\varphi}_y(t, t-i \,|\, f_t, f_{t-i})\right)$. We describe the MC-prediction as follows,

$$\begin{aligned}\hat{f}_t\left(x, y \,\middle|\, f^{ref}_{t-i}, \hat{\varphi}\left(t, t-i \,\middle|\, f_t, f^{ref}_{t-i}\right)\right) \\ = f^{ref}_{t-i}\left(x - \hat{\varphi}_x\left(t, t-i \,\middle|\, f_t, f^{ref}_{t-i}\right), \right. \\ \left. y - \hat{\varphi}_y\left(t, t-i \,\middle|\, f_t, f^{ref}_{t-i}\right)\right)\end{aligned} \quad (13)$$

where f^{ref}_{t-i} is a processed or distorted version of f_{t-i} that serves as a reference frame. A reference frame at time t is defined as

$$f^{ref}_t(x, y) = v_t(x, y) + n^{ref}_t(x, y). \quad (14)$$

Here, according to (5), n^{ref}_t contains w^{ref}_t that expresses the reference frame distortions. For example, real hybrid encoders utilize closed-loop MC-coding by using the reconstructed-from-compression version of f_{t-i}; hence, we can express w^{ref}_t's variance using (9).

We assume that

$$\hat{\varphi}\left(t, t-i \,\middle|\, f_t, f^{ref}_{t-i}\right) \approx \hat{\varphi}(t, t-i \,|\, f_t, f_{t-i}), \quad (15)$$

i.e., compression does not affect ME accuracy significantly. Hence, (13) is modified to

$$\begin{aligned}&\hat{f}_t\left(x, y \,\middle|\, f^{ref}_{t-i}, \hat{\varphi}(t, t-i \,|\, f_t, f_{t-i})\right) \\ &= f^{ref}_{t-i}\left(x - \hat{\varphi}_x(t, t-i \,|\, f_t, f_{t-i}), y - \hat{\varphi}_y(t, t-i \,|\, f_t, f_{t-i})\right).\end{aligned} \quad (16)$$

The ME is approximated using (15); however, the compression still affects the MC residual through $\sigma^2_{w,compression}$ of the reference frame.

We assume that the object from f_t, which its motion is estimated, is contained in the search area in f_{t-i}. Therefore, we model $\hat{\varphi}(t, t-i \,|\, f_t, f_{t-i})$ to have a displacement error $(\Delta x, \Delta y)$ that depends only on the spatial properties of the ME algorithm, e.g., search resolution. Hence, the error excludes any temporal dependency. Specifically,

$$\begin{aligned}\hat{\varphi}_x(t, t-i \,|\, f_t, f_{t-i}) &= \varphi_x(t, t-i) + \Delta x \\ \hat{\varphi}_y(t, t-i \,|\, f_t, f_{t-i}) &= \varphi_y(t, t-i) + \Delta y\end{aligned} \quad (17)$$

Where Δx and Δy are uniformly distributed in a range defined by the accuracy of the ME algorithm. Using (1), (3) and (17) we develop (16) into

$$\begin{aligned}&\hat{f}_t\left(x, y \,\middle|\, f^{ref}_{t-i}, \hat{\varphi}(t, t-i \,|\, f_t, f_{t-i})\right) \\ &= v\left(x - \varphi_x(t, 0) - \Delta x, y - \varphi_y(t, 0) - \Delta y\right) \\ &\quad + n^{ref}_{t-i}\left(x - \hat{\varphi}_x(t, t-i \,|\, f_t, f_{t-i}), \right. \\ &\qquad \left. y - \hat{\varphi}_y(t, t-i \,|\, f_t, f_{t-i})\right).\end{aligned} \quad (18)$$

Here we used the property $\varphi(t, 0) = \varphi(t, t-i) + \varphi(t-i, 0)$ that follows from the definition in (2).

The MC-prediction error of f_t using f_{t-i} as a reference frame is formulated as

$$e_{t|t-i}(x, y) = f_t(x, y) - \hat{f}_t\left(x, y \,\middle|\, f^{ref}_{t-i}, \hat{\varphi}(t, t-i \,|\, f_t, f_{t-i})\right) \quad (19)$$

In the appendix (supplementary material), we describe in detail the calculation of the autocorrelation function of the MC-prediction error. This derivation results in

$$\begin{aligned}R_{e_i}(k, l) &= 2\left(\sigma^2_{\Delta x} + \sigma^2_{\Delta y}\right) \cdot \left[R_v(k, l) + R_{n^{ref}_{t-i}}(k, l)\right] \\ &\quad - \sigma^2_{\Delta x} \cdot \left[R_v(k-1, l) + R_v(k+1, l) \right. \\ &\qquad \left. + R_{n^{ref}_{t-i}}(k-1, l) + R_{n^{ref}_{t-i}}(k+1, l)\right] \\ &\quad - \sigma^2_{\Delta y} \cdot \left[R_v(k, l-1) + R_v(k, l+1) \right. \\ &\qquad \left. + R_{n^{ref}_{t-i}}(k, l-1) + R_{n^{ref}_{t-i}}(k, l+1)\right] \\ &\quad + R_{\Delta n_{t,t-i}}(k, l)\end{aligned} \quad (20)$$

where $R_{\Delta n_{t,t-i}}(k, l)$ is the autocorrelation of the MC noise difference, denoted as $\Delta n_{t_2,t_1}$ for $t_1 < t_2$ and defined as

$$\begin{aligned}&\Delta n_{t_2,t_1}(x, y) \\ &\quad \equiv n_{t_2}(x, y) - n^{ref}_{t_1}\left(x - \varphi_x(t_2, t_1), y - \varphi_y(t_2, t_1)\right)\end{aligned} \quad (21)$$

The autocorrelation of $\Delta n_{t_2,t_1}$ is, from the appendix (supplementary material):

$$R_{\Delta n_{t_2,t_1}}(k, l) = \left[2\sigma^2_q \cdot (t_2 - t_1) + \sigma^2_{w_{t_1}} + \sigma^2_{w_{t_2}}\right] \cdot \delta(k, l) \quad (22)$$

The following explicit form of (20) results in [see the appendix (supplementary material)].

$$\begin{aligned}
&R_{e_i}(k,l)\\
&= 2\left[\sigma_{\Delta x}^2+\sigma_{\Delta y}^2\right]\cdot\left[\sigma_v^2\cdot\rho_v^{|k|+|l|}+\left(L\sigma_q^2+\sigma_{w,ref}^2\right)\cdot\delta(k,l)\right]\\
&\quad-\sigma_{\Delta x}^2\sigma_v^2\rho_v^{|l|}\cdot\left[\rho_v^{|k-1|}+\rho_v^{|k+1|}\right]\\
&\quad-\sigma_{\Delta x}^2\left[L\sigma_q^2+\sigma_{w,ref}^2\right]\cdot[\delta(k-1,l)+\delta(k+1,l)]\\
&\quad-\sigma_{\Delta y}^2\sigma_v^2\rho_v^{|k|}\cdot\left[\rho_v^{|l-1|}+\rho_v^{|l+1|}\right]\\
&\quad-\sigma_{\Delta y}^2\left[L\sigma_q^2+\sigma_{w,ref}^2\right]\cdot[\delta(k,l-1)+\delta(k,l+1)]\\
&\quad+\left[2i\sigma_q^2+\sigma_{w,current}^2+\sigma_{w,ref}^2\right]\cdot\delta(k,l)
\end{aligned}\tag{23}$$

From this the error variance is readily obtained as

$$\begin{aligned}
&R_{e_i}(0,0)\\
&= 2\left(\sigma_{\Delta x}^2+\sigma_{\Delta y}^2\right)\cdot\left[\sigma_v^2\cdot(1-\rho_v)+\left(L\sigma_q^2+\sigma_{w,ref}^2\right)\right]\\
&\quad+2i\sigma_q^2+\sigma_{w,current}^2+\sigma_{w,ref}^2
\end{aligned}\tag{24}$$

The last expression shows a linear relation between the variance and the temporal-distance represented here in frame units, i. Translation of the temporal-distance to seconds (denoted as d_t) is possible using (8).

We considered prediction using a single reference-frame as applied in classical compression standards and systems. Recent standards however also support multiple reference frames, where the encoder selects a reference frame from a set of previously decoded frames. While the performance gain depends on the video content [21], the previous frame is expected to provide the best prediction. The MSE (24) is an expected value of a random variable. Some given set of reference frames will have corresponding realizations of the squared-errors, and obviously there is a low probability that a more distant frame will have a lower squared-error.

B. MC-Prediction of an Absent Frame

Let us consider temporal upsampling by a factor of D using MC-FRUC, i.e., $D-1$ missing frames are interpolated between each two existing frames. The available frames are denoted as f_0 and f_D, and the interpolated frames are denoted as $\left\{\hat{f}_j\right\}_{j=1}^{D-1}$. We consider the interpolation of a block in the j^{th} interpolated frame, where $j\in\{1,\ldots,D-1\}$. The corresponding unavailable frame is denoted as f_j.

The prediction includes estimation of the motion between the j^{th} frame and each of the available frames, f_0 and f_D. The estimation is done using f_0 and f_D. $\hat{\varphi}(j,0\,|f_0,f_D)$ and $\hat{\varphi}(D,j\,|f_0,f_D)$ denote the estimated motion at f_j relative to frames f_0 and f_D, respectively. We assume

$$\begin{aligned}
\hat{\varphi}(j,0\,|f_0,f_D) &= \left(\varphi_x(j,0)+\Delta x_0^{abs},\varphi_y(j,0)+\Delta y_0^{abs}\right)\\
\hat{\varphi}(D,j\,|f_0,f_D) &= \left(\varphi_x(D,j)+\Delta x_D^{abs},\varphi_y(D,j)+\Delta y_D^{abs}\right)
\end{aligned}\tag{25}$$

where Δx_0^{abs} and Δx_D^{abs} are assumed to be independent Gaussian random variables with zero-mean and variance

$$\sigma_{\Delta x^{abs}}^2=\gamma_{abs}\cdot\sigma_{\Delta x}^2\tag{26}$$

where $\sigma_{\Delta x}^2$ is the variance of Δx, which was defined above for the case of an available frame, and $\gamma_{abs}>1$ denotes effect of the absence of the frame on the spatial accuracy of the ME. Δy_0^{abs} and Δy_D^{abs} are similarly defined by replacing x with y. As we suggest (26) as a simple model for $\sigma_{\Delta x^{abs}}^2$, one can adapt the model to a given FRUC method by replacing this formulation (e.g., insert a dependency on j).

The overall prediction is calculated using two prediction signals. The backward prediction defined as

$$\begin{aligned}
&\hat{f}_j\left(x,y\,|f_0,\hat{\varphi}(j,0\,|f_0,f_D)\right)\\
&\quad=f_0\left(x-\hat{\varphi}_x(j,0\,|f_0,f_D),y-\hat{\varphi}_y(j,0\,|f_0,f_D)\right).
\end{aligned}\tag{27}$$

and the forward prediction defined as

$$\begin{aligned}
&\hat{f}_j\left(x,y\,|f_D,\hat{\varphi}(D,j\,|f_0,f_D)\right)\\
&=f_D\left(x+\hat{\varphi}_x(D,j\,|f_0,f_D),y+\hat{\varphi}_y(D,j\,|f_0,f_D)\right).
\end{aligned}\tag{28}$$

The final prediction is achieved by a linear combination of (27) and (28):

$$\begin{aligned}
&\hat{f}_j^{final}(x,y\,|f_0,f_D)\\
&\quad=\theta\cdot\hat{f}_j\left(x,y\,|f_0,\hat{\varphi}(j,0\,|f_0,f_D)\right)\\
&\qquad+[1-\theta]\cdot\hat{f}_j\left(x,y\,|f_D,\hat{\varphi}(D,j\,|f_0,f_D)\right)
\end{aligned}\tag{29}$$

where θ is a weight parameter that adjusts the relative influence of the forward and backward frames, using a value in the range [0, 1]. The prediction error being expressed as

$$e_{j|0,D}^{absent}(x,y)=f_j(x,y)-\hat{f}_j^{final}(x,y\,|f_0,f_D)\tag{30}$$

The appendix (supplementary material) describes in detail the calculation of the autocorrelation function of the MC-prediction error. The result is

$$\begin{aligned}
&R_{e_{j|0,D}^{absent}}(k,l)\\
&=\theta^2\cdot R_{\Delta n_{j,0}}(k,l)+(1-\theta)^2\cdot R_{\Delta n_{D,j}}(k,l)\\
&\quad+\sigma_{\Delta x^{abs}}^2\cdot\left[\theta^2+(1-\theta)^2\right]\\
&\qquad\times[2R_v(k,l)-R_v(k-1,l)-R_v(k+1,l)\\
&\quad+2R_{n_0}(k,l)-R_{n_0}(k-1,l)-R_{n_0}(k+1,l)]\\
&\quad+\sigma_{\Delta y^{abs}}^2\cdot\left[\theta^2+(1-\theta)^2\right]\\
&\qquad\times[2R_v(k,l)-R_v(k,l-1)-R_v(k,l+1)\\
&\quad+2R_{n_0}(k,l)-R_{n_0}(k,l-1)-R_{n_0}(k,l+1)]
\end{aligned}\tag{31}$$

Let us study the variance of the error. This variance is also the mean-squared error (MSE) of the interpolation procedure; hence, it is useful for performance evaluation in applications such as FRUC. Using (4), (6) and (22), we obtain from (31) the following MSE expression.

$$\begin{aligned}
R_{e_{j|0,D}^{absent}}(0,0) &= \theta^2\cdot\left[2\sigma_q^2 j+\sigma_{w_0}^2+\sigma_{w_j}^2\right]\\
&\quad+(1-\theta)^2\cdot\left[2\sigma_q^2(D-j)+\sigma_{w_0}^2+\sigma_{w_j}^2\right]\\
&\quad+2\left(\sigma_{\Delta x^{abs}}^2+\sigma_{\Delta y^{abs}}^2\right)\cdot\left[\theta^2+(1-\theta)^2\right]\\
&\quad\times\left[(1-\rho_v)\cdot\sigma_v^2+L\sigma_q^2+\sigma_{w_0}^2\right]
\end{aligned}\tag{32}$$

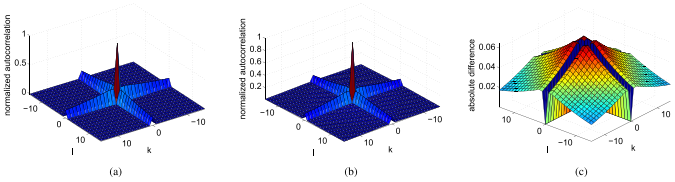


Fig. 2. Estimate of MC-residual autocorrelation in MC-coding. (a) full model (23). (b) simplified using a separable model (34). (c) absolute difference due to simplification.

Usually, θ is set to 0.5 for the central part of the interpolated block. We assume $\theta = 0.5$ for the entire interpolated area; hence, (32) becomes

$$R_{e_{j|0,D}^{absent}}(0,0) = \frac{1}{2}\cdot\left[\sigma_q^2 D + \sigma_{w_0}^2 + \sigma_{w_j}^2\right] + \left(\sigma_{\Delta x^{abs}}^2 + \sigma_{\Delta y^{abs}}^2\right) \cdot\left[(1-\rho_v)\cdot\sigma_v^2 + L\sigma_q^2 + \sigma_{w_0}^2\right]. \quad (33)$$

The last expression shows that the variance is a linear function of the temporal-distance between the available frames, D. Moreover, according to (8), the linear relation with σ_q^2 implies a linear relation with the basic temporal-distance derived from the frame-rate.

FRUC may be applied on processed or reconstructed-from-compression video. The quality of the video affects FRUC performance. Our model supports these cases through the noise components of f_0 and f_D frames; i.e., by including the processed video's MSE in $\sigma_{w_0}^2$ and $\sigma_{w_D}^2$, as in (9) and (12).

C. Simplified Autocorrelation Models for MC-Prediction Error in Coding

In (23) we proposed an autocorrelation function for the error of MC-prediction of an available frame, as in coding applications. Since (23) is rather complicated, it may be useful to have also a simpler autocorrelation model. Here we propose simpler autocorrelation models for the MC-residual in coding systems. The autocorrelation of MC-FRUC can be similarly simplified; however, it is unnecessary since FRUC analysis usually considers only the variance, which is equal to the interpolation MSE.

Similarly to [5] and [11], we construct a model of a separable form from the complicated autocorrelation function. As a result, the linearity of the variance in the temporal-distance is kept.

The variance-normalized autocorrelation function (VNACF) is defined as $\rho_{e_i}(k,l) = \frac{R_{e_i}(k,l)}{R_{e_i}(0,0)}$. The VNACF along the horizontal axis is defined as $\rho_{e_i}^{horz}(k) = \frac{R_{e_i}(k,0)}{R_{e_i}(0,0)}$, and the VNACF along the vertical axis is defined correspondingly and denoted as $\rho_{e_i}^{vert}(l)$. Accordingly, a separable form of $R_{e_i}(k,l)$ is:

$$R_{e_i}^{sep}(k,l) = R_{e_i}(0,0)\cdot\rho_{e_i}^{horz}(k)\,\rho_{e_i}^{vert}(l) \quad (34)$$

Let us derive a separable model in the form of (34) for the autocorrelation function given in (23). This requires the $R_{e_i}(0,0)$ from (24), and the calculation of $\rho_{e_i}^{horz}(k)$ and $\rho_{e_i}^{vert}(l)$. First, we calculate $R_{e_i}(k,0)$ as follows.

$$\begin{aligned} R_{e_i}(k,0) &= 2\left[\sigma_{\Delta x}^2 + \sigma_{\Delta y}^2\right]\cdot\left[\sigma_v^2\cdot\rho_v^{|k|} + \left(L\sigma_q^2 + \sigma_{w,ref}^2\right)\cdot\delta(k)\right] \\ &\quad - \sigma_{\Delta x}^2\sigma_v^2\cdot\left[\rho_v^{|k-1|} + \rho_v^{|k+1|}\right] \\ &\quad - \sigma_{\Delta x}^2\cdot\left(L\sigma_q^2 + \sigma_{w,ref}^2\right)\cdot[\delta(k-1) + \delta(k+1)] \\ &\quad - 2\sigma_{\Delta y}^2\sigma_v^2\rho_v^{|k|+1} + \left(2i\sigma_q^2 + \sigma_{w,current}^2 + \sigma_{w,ref}^2\right)\cdot\delta(k) \end{aligned} \quad (35)$$

Then, we get $\rho_{e_i}^{horz}(k)$ by dividing the last expression by $R_{e_i}(0,0)$ given in (24). $\rho_{e_i}^{vert}(l)$ is achieved similarly by replacing x and k with y and l, respectively. A comparison of the original and simplified autocorrelation (Figs. 2a, 2b,) shows high similarity with very small differences (Fig. 2c).

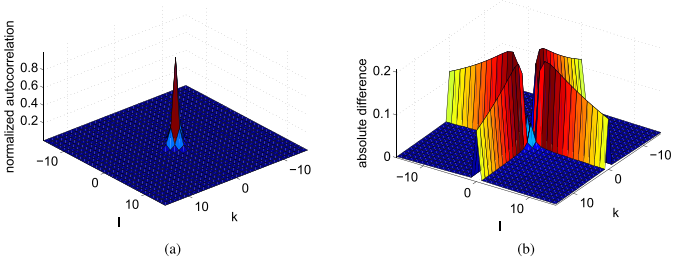


Fig. 3. Simplified autocorrelation function using first-order Markov model. (a) normalized autocorrelation. (b) absolute difference from full model (23).

While the autocorrelation function (23) was simplified to be separable (34), one may benefit from even further simplification of the axis-autocorrelation functions (e.g., (35)). We propose to postulate the autocorrelation function as a separable first-order Markov model. As a result, the horizontal and vertical autocorrelation functions become exponential, i.e.,

$$R_{e_i}^{Markov}(k,l) = R_{e_i}(0,0)\cdot\rho_{h,e_i}^{|k|}\rho_{v,e_i}^{|l|}. \quad (36)$$

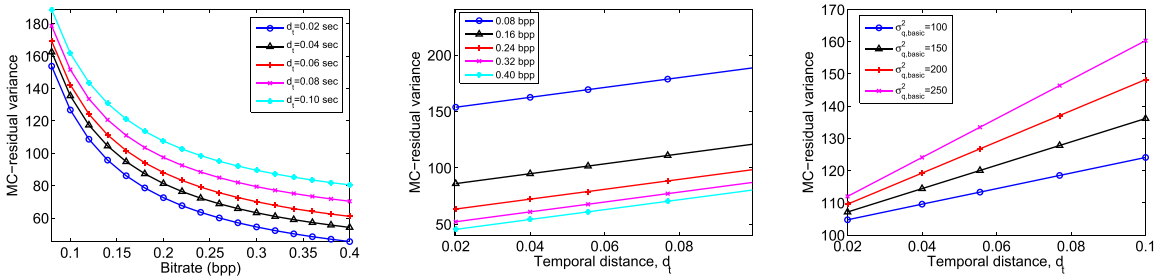
Where R_{e_i} and $R_{e_i}(0,0)$ are the autocorrelation and variance of the accurate model (23). We define the correlation coefficients as follows,

$$\rho_{h,e_i} = \frac{R_{e_i}(1,0)}{R_{e_i}(0,0)} \quad and \quad \rho_{v,e_i} = \frac{R_{e_i}(0,1)}{R_{e_i}(0,0)}. \quad (37)$$

This model differs from the accurate model (23) and the previous simplification (34) in its significantly lower values along the horizontal and vertical axes (Fig. 3a). However, for coordinates that are not on the main axes, the difference from the accurate model is small (Fig. 3b), even more than in the former simplified model (Fig. 2c). In general, we consider this Markov model (36) as an acceptable approximation when further simplifications are needed.

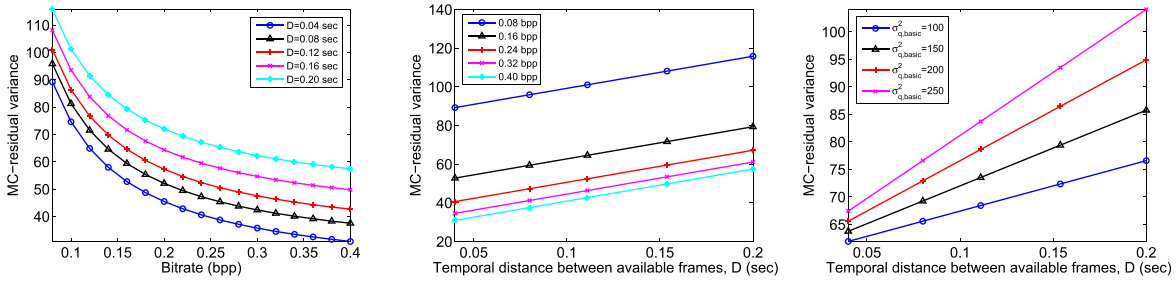
IV. Theoretical Estimates

In this section we explore the model behavior for various signal and compression characteristics. We empirically calculated the frame-statistics of the 'Old town cross' sequence (720×720 at 50fps) and set $\sigma_v^2 = 2300$, $\rho_v = 0.95$, as well as arbitrarily set $L = 5$. The local noise component, σ_w^2, was calculated as follows. $\sigma_{w,basic}^2$ was set to zero, whereas



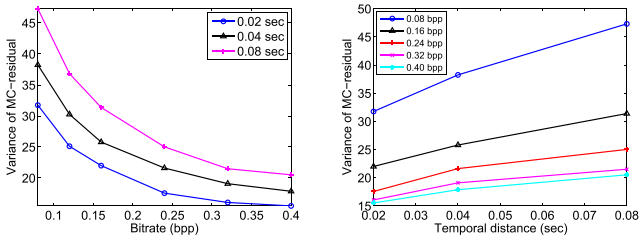
(a) (b) (c)

Fig. 4. Estimate of MC-residual variance in MC-coding. (a) as function of bit-rate for various frame-rates (temporal distances). (b) as function of temporal-distance for various bit-rates. (c) as function of temporal-distance for various motion-energy values $\sigma^2_{q,basic}$.



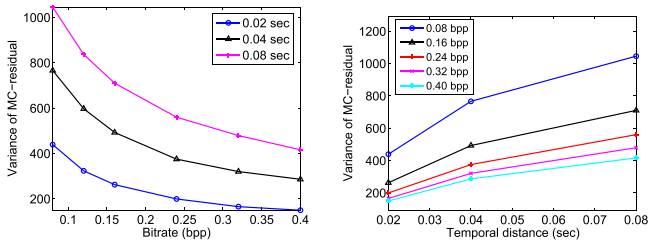
(a) (b) (c)

Fig. 5. Estimation of MC-residual variance in MC-FRUC (i.e., estimation of interpolation MSE). $\gamma_{abs} = 2$. (a) as function of bit-rate for various interpolation factors (temporal distances). (b) as function of interpolation factors (temporal-distance) for various bit-rates. (c) as function of temporal-distance for various motion-energy values $\sigma^2_{q,basic}$.



(a) (b)

Fig. 6. Measured MC-residual statistics in MC-coding of 'Old town cross' sequence (grayscale, frame size 720×720, 10 seconds length). (a) as function of bit-rate for various temporal-distance values. (b) as function of temporal-distance for various bit-rates.



(a) (b)

Fig. 7. Measured MC-residual statistics in MC-coding of 'Parkrun' sequence (grayscale, frame size 720×720, 8 seconds length). (a) as function of bit-rate for various temporal-distance values. (b) as function of temporal-distance for various bit-rates.

$\sigma^2_{w,compression}$ was calculated using (10) with $\alpha = 1$ and $\beta = 10$. We assume ME in half-pel accuracy; therefore, $\Delta x, \Delta y \in [-0.25, 0.25]$ and $\sigma^2_{\Delta x} = \sigma^2_{\Delta y} = (2 \times 0.25)^2/12$.

A. Motion-Compensated Coding

First, we examine the estimated variance as the bit-rate varies (Fig. 4a). The variance is monotonically decreasing as the bit-rate increases, this is due to improved quality of the reference frame that increases its similarity to the coded frame. The convex shape is expected as it is a distortion-rate function.

We assume the reference and the coded frames are adjacent, hence the frame-rate and the temporal-distance can be alternately referred using $d_t = \frac{1}{F_{rate}}$. The estimated variance is linearly increasing as the temporal-distance increases (Fig. 4b). This is justified by the reduced similarity between the reference and coded frames as they get farther.

We compared our estimation for varying motion-complexity of the coded video expressed by $\sigma^2_{q,basic}$ (Fig. 4c). The estimated variance increases together with the motion-complexity. This conforms with the fact that more complex motion degrades the ME results and increases the MC-residual energy.

B. Motion-Compensated Frame-Rate Up Conversion

Let us consider our estimations for the MC-FRUC MSE (33). The equations for the MC-FRUC MSE (33) and the residual variance in MC-coding (24) are similar. Therefore, resembling behavior is expected, and indeed observed in Fig. 5. The explanations given above for MC-coding (see section IV-A) also hold here.

V. Experimental Results

A. Motion-Compensated Coding

We measured the average MC-residual variance in an H.264 software [22] (Baseline profile using constant bit-rate,

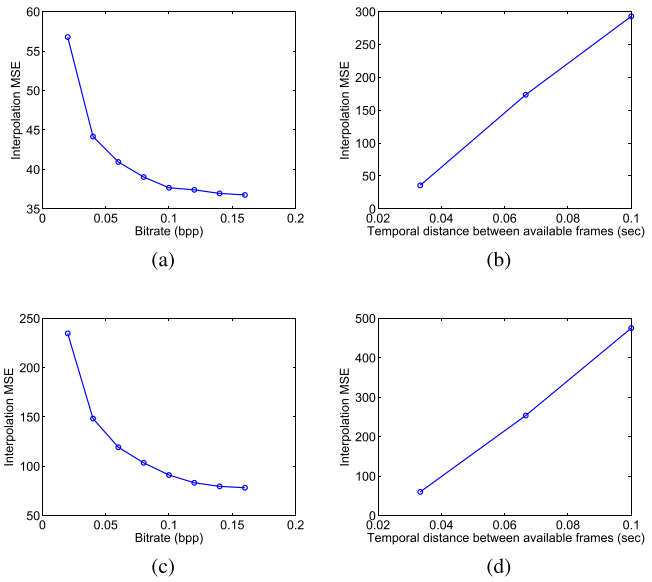


Fig. 8. Measured MSE in MC-FRUC applied on the 'Ice' (a)-(b) and 'Harbour' (c)-(d) sequences (grayscale, frame size 576 × 576, 60fps). (a), (c) as function of bit-rate. (b), (d) as function of temporal-distance (i.e., varying temporal-interpolation factors) for raw video.

full-search ME in a 16-pixel range within the previous frame, and the default allowed block-partitions: intra4×4/16×16, and inter8×8/8×16/16×8/16×16) for the 'Old town cross' and 'Parkrun' sequences (Figs. 6, 7). The variance has a monotonically decreasing convex shape as function of the bit-rate (Figs. 6a, 7a), as in our model (Fig. 4a). In addition, the variance has a relatively linearly-increasing behavior as a function of the temporal-distance (Figs. 6b, 7b), this also conforms with our model estimates (Fig. 4b). We suggest that the small deviation from linearity can be an indirect result of encoding sequences at several frame-rates, as the frame-rate (together with other signal characteristics) affects the encoder decisions (e.g., assignment of a coding-mode and a bit-budget). The 'Parkrun' sequence contains more complex motion than 'Old town cross'; as a result, its residual variance values are significantly higher (Figs. 6a, 7a). This is also expressed in our model (Fig. 4c); hence, the theory qualitatively explains the empirical results.

B. Motion-Compensated Frame-Rate Up Conversion

In section III-B we gave an expression for the MC-FRUC error (33). Here we compare the behavior of the theoretical model with experimental results obtained from an MC-FRUC procedure implemented in Matlab. The variance of MC-prediction error in FRUC equals to the interpolation MSE; hence we here refer to them interchangeably. We examined the dependency of FRUC MSE on temporal-distance and bit-rate. For our experiments, we implemented an MC-FRUC algorithm that applies bidirectional motion-estimation with half-pel accuracy. We considered the central-interpolated frames for upsampling factors $D = 2, 4, 6$ (i.e., $j = \frac{D}{2}$ for even D values). Hence, we studied the relation of the MSE to the temporal-distance by applying FRUC at varying interpolation factors, D, for a fixed frame-rate. The experiments showed an approximately linear increment of the MSE together with the temporal-distance (Figs. 8b, 8d). In addition, its relation to the bit-rate has a convex-decreasing shape (Figs. 8a, 8c). 'Ice' sequence contains more static regions than 'Harbour', i.e., its motion is simpler. Accordingly, higher MSE values are observed for 'Harbour'. The above observations are correspondingly expressed in the theoretical estimates (Fig. 5).

VI. Low Bit-Rate Video Coding: The Advantages of Spatio-Temporal Down-Scaling

Recent video compression standards entail impressive rate-distortion performance. However, as in prior standards, coding at low bit-rates results in reconstructed video with severe artifacts such as blockiness. This poor quality is due to the reduced bit-budget that can be allocated to each block.

It is known (see [20], [23]) that image compression at low bit-rates can be improved by down-scaling the image before compression and up-scaling it to its original size after reconstruction. For a block-based compression method with a fixed block size, a smaller image contains fewer blocks. Therefore, the per-block bit-budget grows as the image gets smaller, and the compression distortion decreases. However, image down-sampling implies removal of high-frequency information; hence, it also reduces the quality. This tradeoff between compression and down-sampling errors makes the down-sampling profitable at low bit-rates. Bruckstein et al. [20] proposed a theoretical explanation for these observations by modeling the JPEG compression standard as a block-based codec that utilizes transform-coding.

While video is a 3D signal, modern hybrid compression methods perform transform-coding on 2D spatial blocks within each frame (usually, after subtracting a corresponding prediction). Hence, the spatial and temporal dimensions of the video affect the number of blocks one has to encode. Consequently, reducing the video dimensions will result in higher bit-budget per each block and therefore smaller compression error. Whereas this is similar to the static image case [20], video compression includes a more complex relation between a block's bit-budget and its compression error. In static image compression, the bit-budget affects only the quantization. However, applying compression on video is a significantly more complex procedure; therefore, the block's bit-budget in video compression has wider effect than just adjusting the quantizer parameters. First to be affected is the chosen coding-mode, i.e., the prediction type (e.g., spatial or temporal). Next to be influenced is the prediction result, since it depends on previously decoded data. Then, the prediction error is transform-coded and quantized according to the bit-budget. Extensions of the scaling-compression approach for video were proposed in [24]–[28], referred to as *down-sampling based video coding*. However, these studies suggested only spatial scaling, whereas the temporal dimension was left untouched.

Temporal resolution reduction for compression at low bit-rates is mainly addressed in studies on frame skipping mechanisms [29]–[33]. While suggestions in [29]–[31] are motivated by technical considerations only, Liu and Kuo [32] and Vetro et al. [33] explain frame-skipping via general

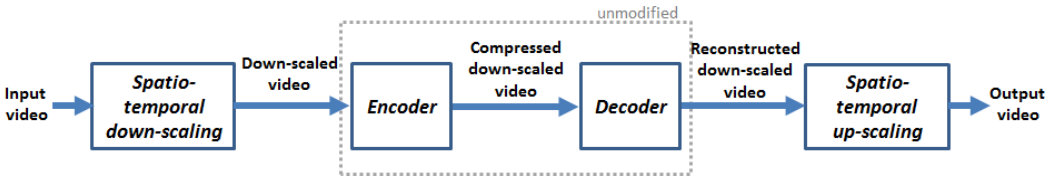


Fig. 9. The structure of the proposed compression-scaling system.

rate-distortion analyses. Song and Kuo [34] proposed a practical rate-control algorithm that balances between spatial and temporal quality, using adaptive frame-rate selection and frame-level bit-allocation; however, no theoretical explanation for its performance was provided.

Many studies limit their scope to rate-distortion analysis without considering the special statistical properties of the video signal (see [28], [33]), or use a quantization-distortion framework where the starting point is at the transform-coefficients stage. Usually relations between the pixel domain and the signal to be transform-coded (e.g., prediction residuals) are separately studied (see [5], [6]). These choices for considering a partial scope of the problem are surely due to the difficulty in providing an accurate mathematical modeling of the video signal and the very complex video compression systems, as discussed in [35]. Here we aim to model theoretically the compression at low bit-rates in a wider scope than usual. Specifically, we provide an elaborate compression model that includes analysis of the coding-mode usage, the motion-compensated prediction and the transform coding. Furthermore, we express the compression distortion as a function of the bit-budget and spatio-temporal properties of the input video in the pixel domain.

In this section, the analysis proposed in [20] for still images is adapted to video signals. A comprehensive spatio-temporal analysis of the compression is proposed, and the optimal spatio-temporal down-scaling factors are examined. We show, analytically and then verify experimentally, that at low bit-rates, we benefit from applying a spatio-temporal down-scaling (i.e., reduction of frame-rate and frame-size, before the compression followed by a corresponding up-scaling, see Fig. 9).

A. Signal Model for Multi-Resolution Analysis

Let us consider a video signal of one-second length. We assume it is defined on the unit cube, and represented by the function $f_v(x,y,t) : [0,1]\times[0,1]\times[0,1]\rightarrow\mathbb{R}$. A set of T frames is defined in the unit cube as

$$\left\{ f_v(x,y,t) \middle| (x,y)\in[0,1]\times[0,1], \quad t\in\{i/T\}_{i=0}^{T-1} \right\} \tag{38}$$

A frame, $f_v(x,y,t=h)$, is assumed to be a realization of a 2D random process. We assume it is wide-sense stationary (WSS) with zero mean in the form of separable first-order Markov model; i.e., the spatial autocorrelation of a frame is

$$R_v(\tau_x,\tau_y) = \sigma_v^2 \cdot e^{-\alpha_x|\tau_x|} e^{-\alpha_y|\tau_y|}. \tag{39}$$

Since we study a block-based compression system, it is useful to consider partitioning of a frame into $M\cdot N$ equal-size 2D-blocks, i.e., the h^{th} frame in the sequence is divided into the following set of 2D-regions defined as:

$$\Delta_{ij}^h \equiv \left[\frac{i-1}{M},\frac{i}{M}\right]\times\left[\frac{j-1}{N},\frac{j}{N}\right] \quad for \quad \begin{matrix} i=1,...,M \\ j=1,...,N \end{matrix} \tag{40}$$

We refer to M and N as the spatial slicing parameters, and to T as the temporal slicing parameter. M, N and T are the slicing parameters for the spatial-horizontal, spatial-vertical and temporal directions, respectively.

We assume block-based compression with a fixed block size denoted as $W_{block}\times H_{block}$; e.g., $W_{block}=H_{block}=16$ for H.264 macroblocks. The block dimensions relate the spatial slicing parameters with the actual frame size according to $W_{frame}=W_{block}\cdot M$ and $H_{frame}=H_{block}\cdot N$. The video considered is one-second in length; hence, the frame-rate and the temporal-slicing parameter obey $F_{rate}=T$.

We define the down-scaling factor as the ratio between dimension values of the original and the down-scaled videos. D_M, D_N and D_T denote the spatial-horizontal, spatial-vertical and temporal down-scaling factors, and hold:

$$D_M = W_{frame}^{original}/W_{frame}^{scaled} = M^{original}/M^{scaled} \tag{41}$$

$$D_N = H_{frame}^{original}/H_{frame}^{scaled} = N^{original}/N^{scaled} \tag{42}$$

$$D_T = F_{rate}^{original}/F_{rate}^{scaled} = T^{original}/T^{scaled}. \tag{43}$$

Our analysis involves compression of videos in different frame rates and sizes. The variety of spatio-temporal resolutions results in diverse amount of slices (or macroblocks) in the videos, and a wide-ranging bit-budget per slice. Therefore, we measure bit-rates in bit-per-slice (denoted as B_{slice}) units. The B_{slice} value is related to the bits-per-second value, B_{second}, using the spatio-temporal slicing parameters (M, N and T) as $B_{slice}=\frac{B_{second}}{M\cdot N\cdot T}$.

B. Coding-Mode Usage at Low Bit-Rates

Modern hybrid block-based compression systems have several coding modes that are chosen blockwise by the encoder. The main difference between coding modes is the prediction method; e.g., inter prediction utilizes information from previously decoded frames, while intra prediction considers only the current frame. H.264's skip mode is an example for a low bit-cost method that offers a simple motion-compensation prediction without any transform coding of the prediction error. Coding-mode selection depends on factors such as bit-rate, signal properties and run-time limitations.

The application considered aims at improving coding at low bit-rates. Accordingly, we can focus on the unique characteristics of coding-mode usage at low bit-rates. In [36], we

present an analysis of coding mode-usage at low bit-rates, including a model based on experimental observations of H.264's baseline profile. We use here the following framework that was considered experimentally in [36]. First, I-frames and intra-coded blocks were found to be negligible at low bit-rates. Second, partitioning of the 16×16 macroblocks was also found as negligible, as a result of the costly representation of the finer block-partitions. Then, the inter-coding and skip modes were found as dominant, and their usage-frequency was found to depend on compression and signal properties such as bit-rate, frame-rate, and motion-complexity. Moreover, the usage-frequency of the inter-coding mode was found to behave as a fractional-linear function of the bit-rate, and a convergence-rate depending on the frame-rate and motion-complexity.

The analytic model for coding-mode assignment is as follows. We assume that the encoder chooses a coding mode (i.e., intra, inter or skip) for each 2D block, Δ_{ij}^{h}, independently of other blocks in the frame. This deviates from a real encoder, where frame types are assigned before the macroblock processing; e.g., many encoders divide the sequence into frame-groups, each begins with an I-frame and the rest are P-frames. Due to the low bit-rate scenario, we further assume that intra coding in P-frames is used rarely, and therefore can be neglected in the rate-distortion analysis. We represent a block's coding mode as a discrete random variable that depends on the bit-rate B_{slice} and the frame-rate F_{rate}:

$$\Delta_{ij}^{h}\ coding\ mode = \begin{cases} inter, & w.p.\ P_{\text{inter}}\left(B_{slice}, F_{rate}\right) \\ skip, & w.p.\ P_{\text{skip}}\left(B_{slice}, F_{rate}\right) \end{cases} \quad (44)$$

where for a given bit-rate, we clearly have that $P_{\text{inter}}\left(B_{slice}\right) + P_{\text{skip}}\left(B_{slice}\right) = 1$.

We claim that as the bit-rate decreases and approaches very low values, more and more blocks are coded in skip mode instead of inter coding. This process is modeled as a linear-fractional function of the bit-rate B_{slice}, and written as:

$$\begin{aligned} P_{inter}\left(B_{slice}, F_{rate}\right) &= \frac{B_{slice}}{c_m\left(F_{rate}\right) \cdot B_{slice} + d_m\left(F_{rate}\right)} \\ P_{skip}\left(B_{slice}, F_{rate}\right) &= 1 - P_{inter}\left(B_{slice}, F_{rate}\right) \end{aligned} \quad (45)$$

Here c_m and d_m control the asymptotic value of the function and the convergence rate, respectively. c_m and d_m are affected by the frame-rate, F_{rate}, and the motion-characteristics of the video, $\tilde{\sigma}_q^2$ (as was defined in (8)):

$$\begin{aligned} c_m\left(F_{rate}\right) &= \frac{100}{P_{inter}^{asymp,\min} + \gamma_c \cdot \frac{\tilde{\sigma}_q^2}{F_{rate}}} \\ d_m\left(F_{rate}\right) &= \gamma_d + \frac{\tilde{\sigma}_q^2}{F_{rate}} \end{aligned} \quad (46)$$

where $P_{inter}^{asymp,\min}$ is the minimal inter-mode percentage (e.g., as in a video with simple motion), and γ_c, γ_d are normalization constants depending on the motion characteristics and the convergence rate, respectively.

When the frame-rate is higher, the motion-compensated prediction residual has reduced energy and inter coding is more advantageous. Therefore, the inter-coding percentage grows with the frame-rate. As the motion is more complex (i.e., higher $\tilde{\sigma}_q^2$), the skip-mode performance degrades, leading to an increased inter-mode usage when the bit-budget is sufficient (i.e., the bit-rate is above a level determined by the trade-off between the overall quality and bit-cost).

C. Motion-Compensated Coding

1) Autocorrelation of MC-Prediction Residual: According to section VI-B, we can study compression at low bit-rates by considering the inter and skip modes only. These two modes rely on motion-compensated prediction. In this section we examine the motion-compensated prediction error, i.e. the MC-prediction residual. Here we use results from sections III-A and III-C and adapt them to the compression-scaling system studied.

Recall the autocorrelation of the MC prediction residual in its simplified form, (36). Here we define

$$\alpha_x = -W \cdot \log\left(\rho_{f_v,x}\right) \quad and \quad \alpha_y = -H \cdot \log\left(\rho_{f_v,y}\right) \quad (47)$$

where W and H are the frame width and height, respectively, in pixels. Then, we rewrite (4) using (47) and get

$$R_{f_r}\left(\tau_x, \tau_y\right) = \sigma_{f_r}^2 \cdot e^{-\alpha_x|\tau_x|} e^{-\alpha_y|\tau_y|}. \quad (48)$$

Unlike in section III, the signal model here is assumed to be continuous to allow multi-resolution analysis. We define the horizontal and vertical pixel widths, denoted as ε_x and ε_y, respectively. For an original frame size of $H_0 \times W_0$, the pixel widths are $\varepsilon_x = 1/W_0$ and $\varepsilon_y = 1/H_0$. Plugging the pixel widths in the expression of the variance (24) yields

$$\begin{aligned} \sigma_{f_r}^2 = 2\left(\frac{\sigma_{\Delta x}^2}{\varepsilon_x^2} + \frac{\sigma_{\Delta y}^2}{\varepsilon_y^2}\right) \cdot \left[\sigma_v^2 \cdot (1 - \rho_v) + \frac{L}{F_{rate}}\tilde{\sigma}_q^2 + \sigma_{w,ref}^2\right] \\ + 2\tilde{\sigma}_q^2 d_t + \sigma_{w,current}^2 + \sigma_{w,ref}^2 \end{aligned} \quad (49)$$

The continuous form of the correlation coefficients (37) is derived similarly.

We here neglect temporally-local noises other than due to compression and spatial down-scaling. The noise energy values of compression and spatial down-scaling are denoted as $\sigma_{compression}^2$ and $\sigma_{spatial-scaling}^2$, respectively. We assume the compression and scaling noise processes to be independent. The coded frame is affected only by the spatial down-scaling, hence

$$\sigma_{w,current}^2 = \sigma_{spatial-scaling}^2. \quad (50)$$

In contrast, the used reference frame is reconstructed from compression. Therefore, it is affected by both compression and scaling noises; their independence yields

$$\sigma_{w,ref}^2 = \sigma_{spatial-scaling}^2 + \sigma_{compression}^2. \quad (51)$$

2) The Effect of Spatio-Temporal Down-Scaling: Our compression-scaling system examines the compression with temporal down-scaling, i.e., frame-rate reduction. Lower frame-rate implies increased temporal-distance between frames, hence, it affects the motion estimation and compensation procedures. The autocorrelation of MC-prediction residual (36) expresses the quality reduction of

ME and MC as the frame-rate gets lower. The variance (49), which is also the prediction-error energy, increases as the frame-rate decreases.

Adding the spatial down-scaling effect, is analyzed as follows. Note that previous work treated spatial down-scaling before compression of image [20] and video ([24], [28]) signals. Unlike [20] and [28], we consider here a predictive coding system; hence, the original signal, which is down-scaled, is not the transform-coded one. In our analysis, the transform-coded signal is the MC-prediction residual that is modeled according its second-order statistics (48); whereas in (49)-(51), we express the effect of spatial down-scaling of the original video as a part of the temporally-local noise signals of the coded and reference frames, as in (50) and (51), respectively. Here we analyze the error introduced by the spatial scaling and calculate $\sigma^2_{spatial-scaling}$ for a given spatial statistics of a video frame and a down-scaling factor. In the appendix (supplementary material), we calculated the error due to spatial down-scaling, and obtained

$$\begin{aligned}&\sigma^2_{spatial-scaling}\\&=\frac{\sigma_v^2}{\pi^2}\left[I\left(\omega^d_{m,x},\omega^0_{m,x},\omega^d_{m,y},\omega^0_{m,y}\right)+I\left(\omega^d_{m,x},\omega^0_{m,x},0,\omega^d_{m,y}\right)\right.\\&\qquad\left.+I\left(0,\omega^d_{m,x},\omega^d_{m,y},\omega^0_{m,y}\right)\right]\end{aligned}\tag{52}$$

Here $\omega^0_{m,x}=2\pi W_0$, $\omega^0_{m,y}=2\pi H_0$, $\omega^d_{m,x}=2\pi W_d$ and $\omega^d_{m,y}=2\pi H_d$ for original and down-scaled frame-sizes of $W_0\times H_0$ and $W_d\times H_d$, respectively; and $I(\cdot,\cdot,\cdot,\cdot)$ is defined as follows

$$\begin{aligned}&I\left(\omega_{x1},\omega_{x2},\omega_{y1},\omega_{y2}\right)=4\tilde{I}\left(\omega_{x1},\omega_{x2},\alpha_x\right)\tilde{I}\left(\omega_{y1},\omega_{y2},\alpha_y\right)\\&where,\ \tilde{I}\left(\omega_1,\omega_2,\alpha\right)=\arctan\left(\left(\omega_2-\omega_1\right)/\left(\alpha+\frac{\omega_1\omega_2}{\alpha}\right)\right).\end{aligned}\tag{53}$$

Note that the MSE (52) increases with the spatial down-scaling factor.

3) MC-Prediction Error in Inter-Coding and Skip Modes: As explained earlier, we consider two MC-based coding modes: inter-coding and skip modes. In inter-coding the prediction is fairly done and the prediction error is transform coded for the reconstruction. Hence, we consider statistical model defined in this section to represent the MC-prediction residual in inter-coding; i.e., $R^{inter}_{f_r}(\tau_x,\tau_y)\equiv R_{f_r}(\tau_x,\tau_y)$, where R_{f_r} was defined in (48).

In contrast, skip mode is a low bit-cost mode. First, it entails inferior MC-prediction by constructing it from its spatial neighbors; then, the prediction error is not transmitted to the decoder. Assuming the former property only, leads us to a proportional-increment in prediction-error energy relative to inter-coding; i.e.,

$$R^{skip}_{f_r}(0,0)\triangleq\gamma\cdot R_{f_r}(0,0)=\gamma\cdot\sigma^2_{f_r}\tag{54}$$

where $\gamma>1$, and $\sigma^2_{f_r}$ was defined in (49). Let us consider a block Δ_{ij} that is encoded in skip mode. The original block information and its reconstruction are denoted as f_v and $\hat{f}^{skip}_v$, respectively. The reconstruction MSE is calculated as follows.

$$\begin{aligned}&E\left\{MSE_{f_v}\left(\Delta_{ij}\right)\right\}\\&=\frac{1}{A\left(\Delta_{ij}\right)}\int\int_{\Delta_{ij}}E\left\{\left(f_v(x,y)-\hat{f}^{skip}_v(x,y)\right)^2\right\}dxdy\\&=\frac{1}{A\left(\Delta_{ij}\right)}A\left(\Delta_{ij}\right)\cdot R^{skip}_{f_r}(0,0)=\gamma\cdot\sigma^2_{f_r},\end{aligned}\tag{55}$$

where $A\left(\Delta_{ij}\right)$ is the area of the block Δ_{ij}. The use of MC in skip mode yields a dependency of this mode's error on frame-rate and bit-rate as expressed by (49).

D. Predictive Coding Analysis

1) Basic Error Expression: Let us consider a 2D-slice, Δ^h_{ij}, in the video. This block is encoded by a block-based predictive-coding technique. The prediction results in the encoder and decoder are identical, and the prediction error is coded with a lossy encoding. Hence, the overall error is the coding error of the prediction-residual. The prediction-residual signal is represented by the function f_r. The residual of the Δ^h_{ij} slice is a function $f_r:\left[\frac{i-1}{M},\frac{i}{M}\right]\times\left[\frac{j-1}{N},\frac{j}{N}\right]\to\mathbb{R}$.

The prediction-residual, f_r, depends on the prediction method (e.g., intra, inter, etc.). In this section, we consider a general f_r signal, and describe it by properties that are assumed to hold for any relevant prediction method. We model f_r as a WSS process with zero mean and autocorrelation function $R_{f_r}\left(\tau_x,\tau_y\right)$. The reconstructed residual is denoted as $\hat{f}_r$. According to [20], the expected reconstruction-MSE of a WSS signal can be calculated from only one slice of it; hence, the expected MSE of f_r is given by:

$$\begin{aligned}E\left\{MSE_{f_r}\right\}&=E\left\{MSE_{f_r}\left(\Delta^h_{ij}\right)\right\}\\&=MN\iint\limits_{\Delta^h_{ij}}E\left\{\left(f_r(x,y)-\hat{f}_r(x,y)\right)^2\right\}dxdy.\end{aligned}$$

2) Transformation of Prediction-Residual: In hybrid compression, the prediction-residual block f_r is transformed and represented using an orthonormal basis of functions. Recent compression standards support prediction and transform in block sizes that may differ. E.g., H.264 support prediction in block sizes between 4×4 to 16×16, whereas the transform is applied on 4×4 or 8×8 blocks [37], [38]. We denote the ratio between the prediction and the transform block sizes as β, and assume it is a positive integer. These assumptions conform with the studied case of low bit-rate video coding using H.264's baseline profile, where the prediction-blocks are assumed to be 16×16 pixels and the transform is applied on 4×4 blocks.

A slice to be encoded is defined on $\left[0,\frac{1}{M}\right]\times\left[0,\frac{1}{N}\right]$; hence, the transformation is applied separately on β^2 equal-sized square sub-slices of this slice; i.e., on

$$\left[\frac{p-1}{\beta M},\frac{p}{\beta M}\right]\times\left[\frac{q-1}{\beta N},\frac{q}{\beta N}\right]\ for\ \begin{array}{l}p=1,...,\beta\\q=1,...,\beta\end{array}\tag{56}$$

Let us denote the (p,q) sub-slice of the (i,j) slice as $\Delta_{ij,pq}$, where $i\in\{1,...,M\}$, $j\in\{1,...,N\}$, $p,q\in\{1,...,\beta\}$.

The residual signal defined on the region of the sub-slice $\Delta_{ij,pq}$ is denoted as the function $f_{r,\Delta_{ij,pq}}(x,y)$.

The transform coding was analyzed in [20], similarly we get here the following MSE expression of a slice:

$$E\left\{MSE^{Q}_{f_r}(\Delta_{11})\right\} = R_{\Delta_{ij}}(0,0) - \beta^2 MN \cdot \sum_{(k,l)\in\Omega}\sum \mathrm{var}\{F_{kl}\}\cdot\left(1-\frac{K}{2^{2b_{kl}}}\right), \quad (57)$$

where F_{kl} is the (k,l) transform coefficient; b_{kl} is the number of bits for representing F_{kl}; and K is the parameter of quantization error approximation.

3) The Case of Inter Coding: H.264 applies an integer transform that approximates DCT. Accordingly, we adapt the transform model from [20] to our scenario and choose the separable cosine basis for transforming the sub-slices in our analysis. The inter prediction autocorrelation (48) has the form of first-order Markov model; hence, the second-moment of the (k,l) coefficient is calculated like in [20], and results in

$$E\left\{F^2_{kl}\right\} = \sigma^2_{e_i}(F_{rate}, B_{slice})\cdot(2-\delta_k)(2-\delta_l) \times\frac{1}{\beta^2 MN}\cdot Y\left(\frac{\alpha_x}{\beta M};k,k\right)\cdot Y\left(\frac{\alpha_y}{\beta N};l,l\right), \quad (58)$$

where, as defined in [20],

$$Y(A;k,l) \triangleq \int_0^1\int_0^1 e^{-A|x-\xi|}\cdot\cos(k\pi x)\cos(l\pi\xi)dxd\xi.$$

E. Overall Compression

1) Bit-Allocation: Practical transform-coding systems usually have an a-priori bit-allocation rule for dividing a given bit-budget among the transform coefficients. H.264's baseline profile applies uniform quantization on its 4×4 transform coefficients; whereas in high profiles a weighted-quantization (i.e., non-uniform) is carried out on 8×48 transform coefficients.

The a-priori bit-allocation among transform coefficients is modeled by relative bit-allocation. This is similar to the image compression model in [20]; however, we present here several adaptations to treat the joint use of inter and skip modes. Let us define the normalized relative bit-budget of the (k,l) coefficient, which is denoted as $\tilde{Q}_{weight}(k,l)$, its value being in the range [0, 1]. Let us derive the number of bits allocated to transform coefficients of a slice. For compactness of representation, we omit the explicit notation of the coding mode probabilities and write them as P_{inter} and P_{skip}. The total number of slices in the video is $S_{total} = N\cdot M\cdot T$. In addition, the number of inter and skipped slices are calculated as $S_{inter} = P_{inter}\cdot S_{total}$ and $S_{skip} = P_{skip}\cdot S_{total}$, respectively.

We exclude from our analysis some elements, that have opposite effects on the bit-cost. Motion-vectors and coding-mode information are disregarded; on the other hand, the entropy coding is also excluded from our scope. We assume these untreated elements balance their overall effect on the total bit-cost. Furthermore, their indirect effect on the distortion is considered through our proposed models.

The bit-budget for the coefficients of an inter-coded slice is

$$B^{slice}_{coeffs} = \frac{B_{total}}{S_{\text{inter}}} = \frac{B_{total}}{S_{total}\cdot P_{\text{inter}}} = \frac{B_{total}}{M\cdot N\cdot T\cdot P_{\text{inter}}} \quad (59)$$

Each slice consists of β^2 separately-transformed sub-slices. Therefore, we are interested in the bit-budget for the transform coefficients of a sub-slice: $B^{sub-slice}_{coeffs} = \frac{1}{\beta^2}\cdot B^{slice}_{coeffs}$. The number of bits allocated for the (k,l) coefficient as function of the slicing parameters M, N and T is

$$b_{kl} = \tilde{Q}_{weight}(k,l)\cdot B^{sub-slice}_{coeffs} \quad (60)$$

H.264's baseline profile utilizes uniform quantization and a 4×4 transform, therefore $\tilde{Q}_{weight}(k,l) = \frac{1}{16}$ for $1\leqslant k,l\leqslant 4$.

F. The Compression-Scaling System

1) Overall Compression Distortion: As discussed in previous sections, H.264 utilizes three macroblock coding modes: intra, inter and skip. In section VI-B, we modeled the coding-mode usage as probabilities varying with the bit-rate while neglecting usage of intra-coding (44)-(46). Moreover, we analyzed the distortion-rate behavior while the bit-cost of elements such as motion-vectors and coding-mode is considered indirectly by modeling the properties of the transform-coded signal as function of the total bit-rate.

As was shown in [20], the expected MSE of the entire signal reconstruction equals to the expected MSE of a slice, i.e., $E\left[\varepsilon^2_v\right] = E\left[MSE_{f_v}(\Delta^1_{11})\right]$. However, the slice coding-mode affects the resulting reconstruction error. Moreover, the chosen coding-mode is a random-variable with a distribution function given by (44). Hence, we denote the MSE for a given coding-mode as $E\left[MSE_{f_v}(\Delta^1_{11})\middle|\, coding\ mode\right]$. Applying the law of total expectation for the calculation of the expected MSE of a slice yields

$$\begin{aligned} E\left[\varepsilon^2_v\right] &= E\left[MSE_{f_v}\left(\Delta^1_{11}\right)\right] \\ &= E\left[E\left[MSE_{f_v}\left(\Delta^1_{11}\right)\middle|\, coding\ mode\right]\right] \\ &= P_{inter}(B_{slice})\cdot E\left[MSE_{f_v}\left(\Delta^1_{11}\right)\middle|\, inter\ coding\right] \\ &\quad + P_{skip}(B_{slice})\cdot E\left[MSE_{f_v}\left(\Delta^1_{11}\right)\middle|\, skip\ mode\right]. \end{aligned} \quad (61)$$

2) Analysis of the Overall Compression-Scaling System: Recall the structure of the investigated system for improved low bit-rate video coding (Fig. 9). This system suggests to compress a down-scaled video and to up-scale it to its original dimensions after decoding. The compression error of the down-scaled video was studied here resulting in the expression (61) for the error. The error of temporal interpolation by MC-FRUC techniques is given by setting $\sigma^2_{w_0} = MSE_{compression}$, where $MSE_{compression}$ equals

to $E\left[\varepsilon_v^2\right]$ from (61); consequently, (33) is updated to

$$\begin{aligned} &MSE_{FRUC}\left(D_T, j, MSE_{compression}\right) \\ &\quad = \frac{1}{2}\cdot\left[\tilde{\sigma}_q^2\frac{D_T}{F_{rate}} + MSE_{compression} + \sigma_{w_j}^2\right] \\ &\qquad + \left(\sigma_{\Delta x^{abs}}^2 + \sigma_{\Delta y^{abs}}^2\right)\left[(1-\rho_v)\,\sigma_v^2 + \frac{L\tilde{\sigma}_q^2}{F_{rate}}\right. \\ &\qquad\qquad \left. + MSE_{compression}\right] \end{aligned} \tag{62}$$

Here D_T is the frame-rate upsampling factor, defined according to (43) as $D_T = \frac{original\ \ frame\ rate}{T}$, T being the temporal slicing factor. Recall that the spatial down-scaling error is included in the compression error through its effect on the MC-prediction residual.

The spatio-temporal down-scaling operations are applied sequentially, decreasing the frame size and lowering the frame rate by discarding frames (assuming the temporal down-scaling factor is an integer). The down-scaling operation order is important only for computational efficiency, since reducing frame size of omitted frames is unnecessary. In contrast, the operation order of up-scaling after decoding is important for the quality of the result. Our proposed system includes MC-FRUC algorithm for frame interpolation; hence, motion-estimation is performed and its performance affects the interpolation quality. Commonly, motion-estimation and compensation is performed on finer spatial resolution (e.g., half-pel or quarter-pel) where the reference frames are temporarily enlarged for better results [39]. Therefore, spatial up-scaling before applying FRUC gives better results than the opposite order.

The output video consists of two frame types. Firstly, frames that were encoded in the down-scaled video. These frames were spatially down-scaled and up-scaled, before and after compression, respectively; Therefore, the spatial scaling affects them directly, whereas the temporal scaling affects them only indirectly through the lower frame rate in the actually compressed video. The second frame type is the omitted frames that were interpolated after decoding. These frames are affected directly by the temporal scaling, while the spatial scaling affects them indirectly through its distortion on the frames of the first type that are used for the FRUC. The frame types are arranged periodically according to the chosen temporal down-scaling factor.

Since the frames are reconstructed from compressed data or by temporal-interpolation from it, the overall MSE of the output video is a weighted-average of the compression and interpolation errors. The overall MSE is given by

$$\begin{aligned} &MSE_{overall}\,(M,N,T,B) \\ &\quad = \frac{1}{D_T}MSE_{spatial}\,(M,N,T,B) \\ &\qquad + \frac{D_T-1}{D_T}MSE_{spatio-temporal}\,(M,N,T,B). \end{aligned} \tag{63}$$

Where, D_T was defined after (62), and $MSE_{spatial}$ is the MSE of a compressed frame that was only spatially scaled, it is defined as $MSE_{spatial}\,(M,N,T,B) = MSE_{compression}\,(M,N,T,B)$. Note that the temporal-scaling affects these frames indirectly through the statistics of the MC-prediction residual that is spatially coded. $MSE_{spatio-temporal}$ is the MSE of a frame that was discarded in temporal down-scaling procedure. This frame is reconstructed using frames that were only spatially down-scaled; therefore, we set $\sigma_{w_0}^2 = MSE_{spatial}\,(M,N,T,B)$ and define $MSE_{spatio-temporal}$ as the average MSE of interpolated frames:

$$\begin{aligned} &MSE_{spatio-temporal}\,(M,N,T,B) \\ &= \frac{1}{D_T-1}\times\sum_{j=1}^{D_T-1}MSE_{FRUC}\left(T, j, MSE_{spatial}\,(M,N,T,B)\right), \end{aligned} \tag{64}$$

where $MSE_{spatial}$ (that equals to $MSE_{compression}$ (61)) and MSE_{FRUC} (62) are the MSE of the frames of the compressed down-scaled video, and the temporally-interpolated frames, respectively.

G. Results

1) Theoretical Predictions of the Model: We examine the overall compression-scaling system by estimating PSNR for various down-scaling factors at varying bit-rates. We considered the following cases of scaling: only-temporal (Fig. 10a), only-spatial (Fig. 10b), and joint spatio-temporal1 (Fig. 11).

The cases of scaling only in one dimension-type (i.e., temporal or spatial) share the following general behavior. First, we got a pattern of decision regions (Figs. 10a, 10b), where a decision region corresponding to a higher down-scaling factor is located in a lower bit-rate range. Second, as the dimension-related features of the video (i.e., motion and texture in the temporal and spatial dimensions, respectively) become more complex, the estimated PSNR is lower [36]. Moreover, the intersection between scaling-curves occurs at lower bit-rates; hence, the advised down-scaling factor is lower. These estimates are justified by the higher distortion expected from the interpolation due to unrecoverable information when the complexity of the dimension-related-features increases.

Let us analyze further the system behavior for 1D scaling and exemplify it for the spatial dimension and a varying texture level. As the video contains larger amount or higher complexity textures its pixel's variance increases and the correlation coefficient decreases. Our estimations show that compression of a more textured video results in a lower quality (Fig. 10c), since representation of textured images require higher bit-budget. Moreover, texture information resides in high frequency components that are removed in the spatial down-scaling; hence, lower spatial down-scaling factors are preferable for videos with increased texture content (Fig. 10c).

Let us examine the joint spatio-temporal down-scaling. As in the 1D case, any down-scaling of non-trivial signal will introduce information loss and distortions. However, for a given bit-rate and spatio-temporal characteristics of a video, the optimal choice of spatio-temporal down-scaling factors depends on the relation

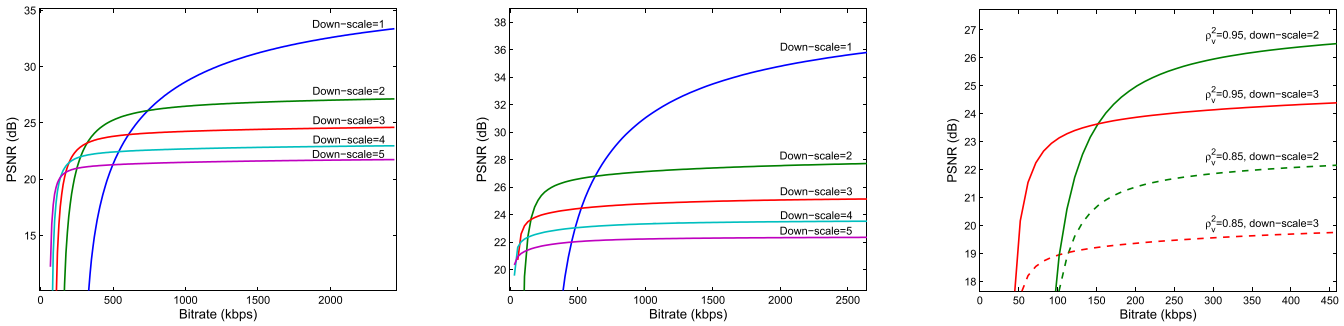


Fig. 10. Theoretical estimates of the overall compression-scaling system PSNR for a typical video ($\sigma_v^2 = 2300$, $\rho_v = 0.95$, $\tilde{\sigma}_q^2 = 250$, $L = 100$). (a) temporal-scaling, (b) spatial-scaling, (c) comparison of spatial-scaling for different texture complexities ($\rho_v = 0.95$ and 0.85).

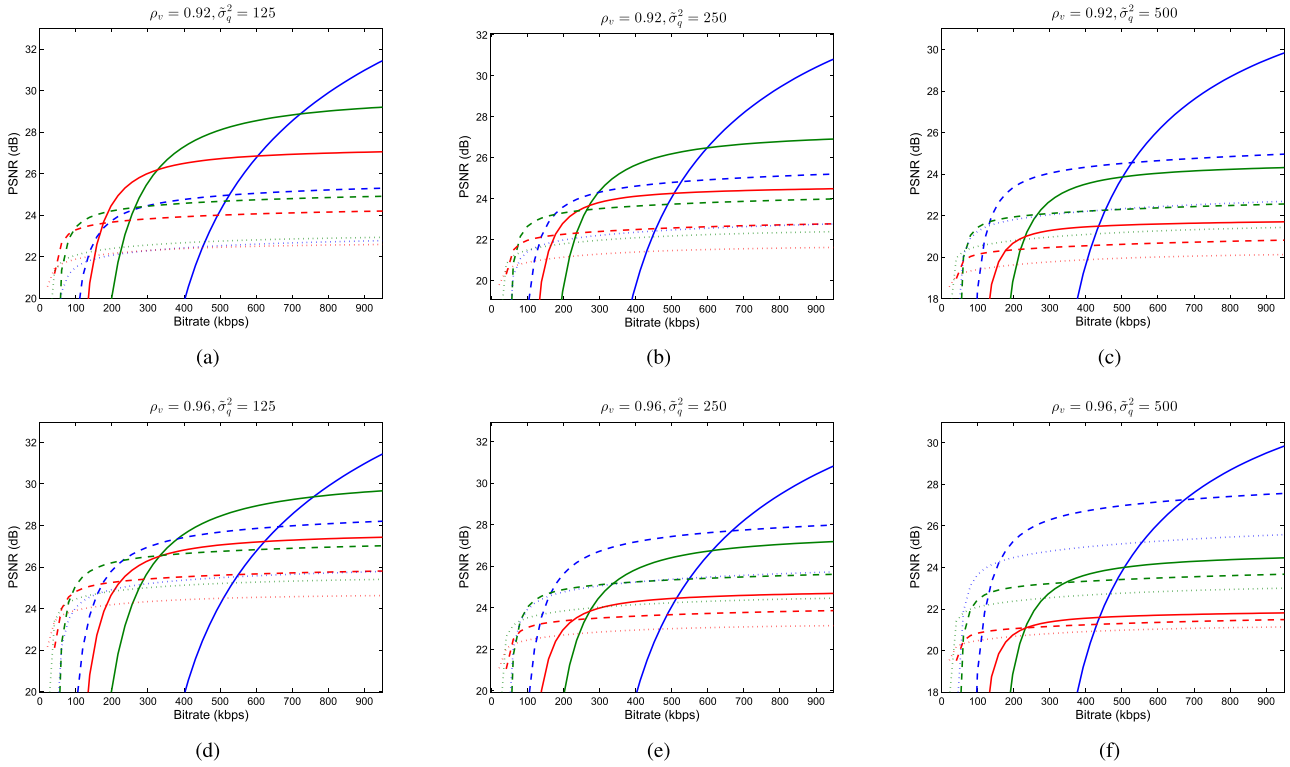
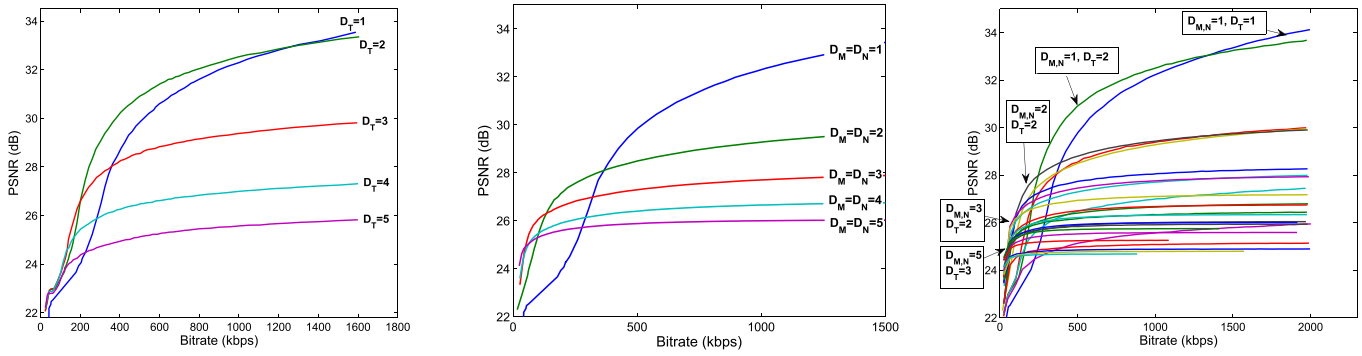


Fig. 11. Theoretical estimation of compression PSNR for spatio-temporal scaling of a video signal with varying texture and motion levels (set using the correlation coefficient ρ_v, and the motion-complexity $\tilde{\sigma}_q^2$, respectively). Spatial down-scale factor is represented by line style: $D_M = D_N = 1$ (solid), 2 (dashed), 3 (dotted). Temporal down-scale is represented by line color: $D_T = 1$ (blue), 2 (green), 3 (red). (Fixed values: $\sigma_v^2 = 2300$ and $L = 100$).

between the complexities of texture and motion. Figure 11 shows PSNR plots for varying levels of motion and texture complexities. Figures 11a-11f are ordered as follows. As the figure located more right, than the motion-complexity, $\tilde{\sigma}_q^2$, is higher. Additionally, as the figure location is lower, than the texture-complexity is lower and ρ_v is higher. Each plot include 9 PSNR curves for combinations of spatio-temporal down-scaling factors, where $D_T \in \{1, 2, 3\}$ and $D_M = D_N \in \{1, 2, 3\}$. The intersections among the PSNR curves define the estimated optimal decision regions. Fig. 11 shows that for higher motion-complexity (i.e. more right sub-figure), then spatial down-scaling is more beneficial than reducing the frame-rate. In addition, higher texture complexity, i.e. upper sub-figure location, makes the temporal down-scaling more preferable. Many optimal combinations of down-scaling factors hold $D_T > 1$ and $D_M > 1$ (Fig. 11). Hence, in these cases both dimensions are down-scaled, implying better results than can be achieved in the 1D scaling systems.

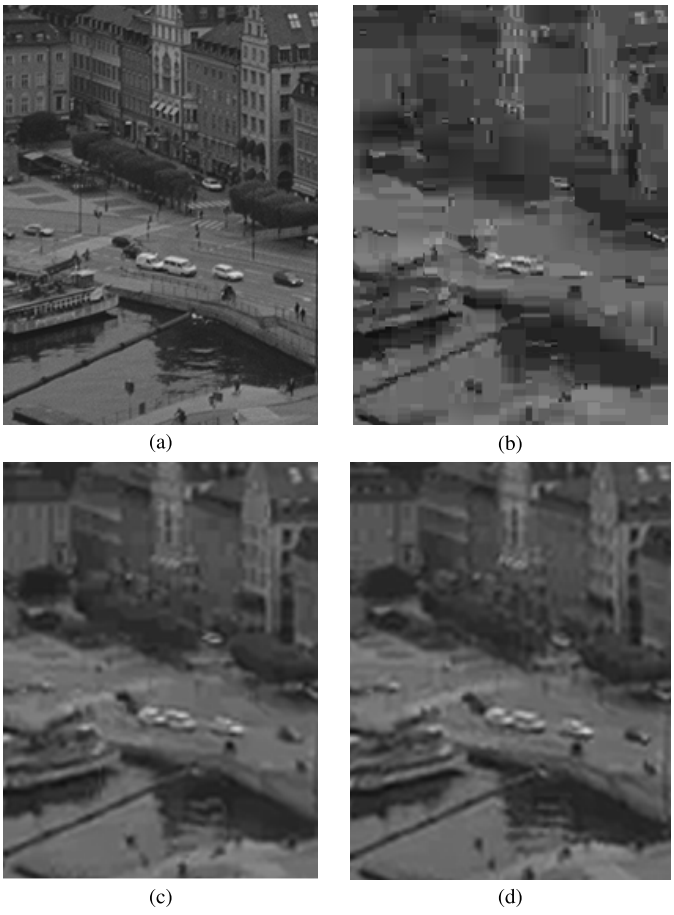
2) Experimental Results: We overview the experimental results of our compression-scaling system, consisting of an H.264 codec [22] (settings as specified in section V-A) and a Matlab implementation of spatio-temporal scaling (including MC-FRUC). We show here results for the 'Old town cross' sequence for temporal-only, spatial-only and joint spatio-temporal scaling (Figures 12a, 12b and 12c, respectively). In [36] we show additional results for the 'Parkrun' sequence that represents a more complex motion.

The experimental results exhibit very nice qualitative agreement with our theoretical model. Specifically, the



(a) (b) (c)

Fig. 12. PSNR of compression-scaling system of 'Old town cross' (720×720, 50fps, grayscale) for (a) temporal, (b) spatial, and (c) spatio-temporal scaling.



(a) (b) (c) (d)

Fig. 13. Demonstration of video compression at low bit-rates. Part of a frame from 'Old town cross' (720×720, 50fps). (a) original, (b) directly compressed at 180kbps, (c) spatial down-sampling by 2 before compression at 180kbps, and (d) spatio-temporal down-sampling by 2 before compression at 180kbps. Larger parts of the frame are shown in [36].

experiments showed the decision-region's pattern and its dependence on the bit-rate and motion/texture complexities (Figures 12a, 12b and 12c), as observed in theoretical predictions (Figures 10a, 10b and 11)).

Let us exemplify the achieved improvement by our system on 'Old town cross'. Firstly, temporal-only scaling (Fig. 12a) showed a PSNR improvement of 2.6dB at 180kbps by reducing the frame-rate in a factor of 3; additionally, bit-savings of 34% were achieved for fixed PSNR of 27dB by halving the frame-rate. Secondly, spatial scaling (Fig. 12b) showed a PSNR improvement of 3.3dB at 180kbps, and bit-savings of 43% at 27dB both by halving the frame width and height (see Fig. 13c for visual demonstration). Finally, the joint spatio-temporal scaling showed the highest improvement by halving the frame-rate, frame width and height that yielded a PSNR gain of 3.9dB at 180kbps, and bit-savings of 56% at fixed PSNR of 27dB (Fig. 12c). These improvements, that are also visually noticeable (Fig. 13d), clearly exceed those of 1D scaling.

VII. Conclusion

The motion-compensation (MC) procedure was studied in this work. Both cases of predicting available and absent frames were theoretically examined, and expressions for the prediction error and its autocorrelation were given. The procedures considered represent the applications of MC in coding and frame-rate up-conversion (FRUC). The analysis is based on a statistical model for the video signal that was presented at the beginning of this paper. Along this study, a special focus was given to the effects of frame-rate and bit-rate on the MC-prediction error. The MC applications in coding and FRUC were studied in the same theoretic framework. Hence, the applications can be compared easily, as the similarities and differences become apparent. For the application of MC-coding, we presented three autocorrelation models at different levels of analytic complexity, which are useful for examination of intricate systems that include MC-coding. As a natural application, we utilized our models and comprehensively analyzed a system that combines compression and spatio-temporal scaling for improving low bit-rate video coding.

References

[1] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.

[2] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

[3] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multihypothesis prediction for motion-compensated video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 957–969, Nov. 2002.

[4] F. Kamisli and J. S. Lim, "Transforms for the motion compensation residual," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2009, pp. 789–792.

[5] K.-C. Hui and W.-C. Siu, "Extended analysis of motion-compensated frame difference for block-based motion prediction error," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1232–1245, May 2007.

[6] W. Niehsen and M. Brunig, "Covariance analysis of motion-compensated frame differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 536–539, Jun. 1999.

[7] C.-F. Chen and K. K. Pang, "The optimal transform of motion-compensated frame difference images in a hybrid coder," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 40, no. 6, pp. 393–397, Jun. 1993.

[8] H. J. Leu, S.-D. Kim, and W.-J. Kim, "Statistical modeling of inter-frame prediction error and its adaptive transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 519–523, Apr. 2011.

[9] W. Zheng, Y. Shishikui, M. Naemura, Y. Kanatsugu, and S. Itoh, "Analysis of space-dependent characteristics of motion-compensated frame differences based on a statistical motion distribution model," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 377–386, Apr. 2002.

[10] L. Guo, O. C. Au, M. Ma, Z. Liang, and P. H. W. Wong, "A novel analytic quantization-distortion model for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 627–641, May 2009.

[11] C.-F. Chen and K. K. Pang, "Hybrid coders with motion compensation," *Multidimensional Syst. Signal Process.*, vol. 3, nos. 2–3, pp. 241–266, 1992.

[12] B. Tao and M. T. Orchard, "Prediction of second-order statistics in motion-compensated video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3. Oct. 1998, pp. 910–914.

[13] R. Feghali, F. Speranza, D. Wang, and A. Vincent, "Video quality metric for bit rate control via joint adjustment of quantization and frame rate," *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 441–446, Mar. 2007.

[14] Y. Zhang, D. Zhao, S. Ma, R. Wang, and W. Gao, "A motion-aligned auto-regressive model for frame rate up conversion," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1248–1258, May 2010.

[15] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 5. May 2005, pp. 4927–4930.

[16] T. Q. Vinh, Y.-C. Kim, and S.-H. Hong, "Frame rate up-conversion using forward-backward jointing motion estimation and spatio-temporal motion vector smoothing," in *Proc. Int. Conf. Comput. Eng. Syst.*, Dec. 2009, pp. 605–609.

[17] D. Wang, L. Zhang, and A. Vincent, "Motion-compensated frame rate up-conversion—Part I: Fast multi-frame motion estimation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 133–141, Jun. 2010.

[18] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion—Part II: New algorithms for frame interpolation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 142–149, Jun. 2010.

[19] G. Dane and T. Q. Nguyen, "Optimal temporal interpolation filter for motion-compensated frame rate up conversion," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 978–991, Apr. 2006.

[20] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-scaling for better transform compression," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1144, Sep. 2003.

[21] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," in *Proc. 12th Int. Packet Video Workshop*, 2002, pp. 1–12.

[22] *x264*. [Online]. Available: http://www.videolan.org/developers/x264.html, accessed Mar. 2012.

[23] X. Wu, X. Zhang, and X. Wang, "Low bit-rate image compression via adaptive down-sampling and constrained least squares upconversion," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 552–561, Mar. 2009.

[24] A. Segall, M. Elad, P. Milanfar, R. Webb, and C. Fogg, "Improved high-definition video by encoding at an intermediate resolution," *Proc. SPIE*, vol. 5308, pp. 1007–1018, Jan. 2004.

[25] V.-A. Nguyen, Y.-P. Tan, and W. Lin, "Adaptive downsampling/upsampling for better video compression at low bit rate," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 1624–1627.

[26] D. Barreto, L. D. Alvarez, R. Molina, A. K. Katsaggelos, and G. M. Callicó, "Region-based super-resolution for compression," *Multidimensional Syst. Signal Process.*, vol. 18, nos. 2–3, pp. 59–81, 2007.

[27] M. Shen, P. Xue, and C. Wang, "Down-sampling based video coding using super-resolution technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 755–765, Jun. 2011.

[28] J. Dong and Y. Ye, "Adaptive downsampling for high-definition video coding," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 2925–2928.

[29] Z. Cui and X. Zhu, "SSIM-based content adaptive frame skipping for low bit rate H.264 video coding," in *Proc. 12th IEEE Int. Conf. Commun. Technol.*, Nov. 2010, pp. 484–487.

[30] F. Pan, Z. P. Lin, X. Lin, S. Rahardja, W. Juwono, and F. Slamet, "Adaptive frame skipping based on spatio-temporal complexity for low bit-rate video coding," *J. Vis. Commun. Image Represent.*, vol. 17, no. 3, pp. 554–563, Jun. 2006.

[31] P. Usach, J. Sastre, and J. M. Lopez, "Variable frame rate and GOP size H.264 rate control for mobile communications," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2009, pp. 1772–1775.

[32] S. Liu and C.-C. J. Kuo, "Joint temporal-spatial bit allocation for video coding with dependency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 15–26, Jan. 2005.

[33] A. Vetro, Y. Wang, and H. Sun, "Rate-distortion optimized video coding considering frameskip," in *Proc. Int. Conf. Image Process.*, vol. 3. 2001, pp. 534–537.

[34] H. Song and C.-C. J. Kuo, "Rate control for low-bit-rate video via variable-encoding frame rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 512–521, Apr. 2001.

[35] Z. He and S. K. Mitra, "From rate-distortion analysis to resource-distortion analysis," *IEEE Circuits Syst. Mag.*, vol. 5, no. 3, pp. 6–18, Sep. 2005.

[36] Y. Dar and A. M. Bruckstein. (Apr. 2014). "Improving low bit-rate video coding using spatio-temporal down-scaling." [Online]. Available: http://arxiv.org/abs/1404.4026

[37] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[38] D. Marpe, T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC fidelity range extensions: Tools, profiles, performance, and application areas," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. Sep. 2005, pp. I-593–I-596.

[39] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. Chichester, U.K.: Wiley, 2003.

Yehuda Dar received the B.Sc. degree in computer engineering and the M.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 2008 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His research interests are in image and video processing, in particular, signal compression and video coding.

Alfred M. Bruckstein received the B.Sc. and M.Sc. degrees from the Technion—Israel Institute of Technology (Technion), Haifa, in 1976 and 1980, respectively, and the Ph.D. degree in electrical engineering from Stanford University, CA. Since 1984, he has been with Technion, where he holds the Ollendorff Chair in Science. His current research interests are in swarm/ant robotics, image and signal processing, analysis and synthesis, pattern recognition, and various aspects of applied geometry.