Exploiting Long-Term Redundancies in Reconstructed Video

Aleksandar Stojanovic and Jens-Rainer Ohm, *Member, IEEE*

***Abstract*—In this paper we propose an enhancement to HEVC that allows the usage of homographies for motion compensation between temporally far distant frames. First, we show the benefit of using homographies compared to block-based motion compensation, particularly for long-term motion compensation in the presence of zoom. Then we show that beyond the case where content is temporally repeated, long-term global motion compensation leads to a better compression ratio in the presence of zooming out. We show that this is due to the change in scale on one hand, and the effects of compression on the reference pictures on the other. Using a model for the distortion of images caused by compression, it will be shown that a finer scale reconstructed image that is warped into the perspective of the image to be encoded can constitute a better predictor than an immediately neighboring picture. We compared the HM software with an enhanced system using long-term global motion compensation with homographies and found that the enhanced system achieves Bjøntegaard Delta Rate savings of up to 38%. With the mechanism behind the gains unveiled, we finally try to find an optimal scale ratio between the frame to be encoded and the reference frame.**

***Index Terms*—Video Coding, HEVC, Global Motion Compensation, Distortion, Scale, Homography**

I. Introduction

THE main principle of video coding is to exploit the temporal and spatial correlation of image sequences for bitrate reduction. In this spirit, we propose a method that uses long-term redundancies in video. To do so we compute homographies which are then encoded into the bitstream. This is currently not supported in HEVC and in this paper we want to show in which context this constitutes an advantage.

A. *Related work*

Sprite coding is a concept that was introduced some time ago in video coding in the context of MPEG-4 object based coding. The idea is that video sequences can be separated in three main parts: static background, camera motion, and moving foreground. Hence, the coding procedure consists in coding a sprite, that is an image containing the entire background scene, and the camera motion, which are then used appropriately for warping the background into every frame of the sequence. Foreground is coded separately, using e.g. motion compensation. Sprite coding and global motion compensation are incorporated in the MPEG-4 standard. A potential advantage of the method is that background can be assumed as less important for high quality coding, and errors stemming from interpolation that naturally occur with camera pan and zoom may not need extra compensation, unlike in hybrid video compression like MPEG-AVC or HEVC.

In the case of the aforementioned sprite coding techniques, a serious challenge is the proper representation of the camera motion, which maps the sprite to the background of each frame of the sequence and vice-versa. An assumption that is used and works relatively well is that camera motion can be represented using an affine motion model. This model is a simplification of the model using homographies used in this paper in the sense that it does not account for perspective effects.

A reference often cited in this context is [1], where a model going beyond an affine motion model is introduced for sprite coding, and along with this a framework for the computation of camera motion is presented. Another idea on global motion compensation is given by background mosaicking. Here in the context of hybrid video compression, all background content visible at any time in the sequence is stored in a separate picture buffer, so that it can be re-used at any time, as presented in [2]. This method was used in MPEG-4 part 2, however lost its importance and was disregarded in MPEG-4 part 10. The idea was used again in [3], where affine motion compensation is used for motion prediction using multiple reference frames. Similar ideas are expressed in [4], however the idea of affine motion was dropped in favor of conventional block-based coding, which simplifies the idea to long-term reference frame buffers, which can be used in H.264/AVC since syntax to move some frames in a long-term reference buffer is provided.

These problems notwithstanding, foreground/background separation was further used in video compression in some applications, namely in the field of perceptive video compression, see [5], [6], or [7], to name only a few, or, earlier, in an application similar in spirit to sprite coding, which was presented in [8]. In this work, the main idea was again that optimal PSNR in background portions of the sequence is not necessary. Another related approach was specifically developed for navigation video sequences, see [9] and [10]. In this type of video rotational background motion is predominant, and is dealt with by using the affine motion model.

During the standardization process of HEVC, higher order motion models have again become relevant. The context here however is not perceptional video coding or separate foreground/background coding, but the classical rate-distortion optimized video compression. In [11], the usage of an affine motion model along with new interpolation techniques results

Aleksandar Stojanovic is with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg (e-mail: aleksandar.stojanovic@uni.lu). Jens-Rainer Ohm is with Institut für Nachrichtentechnik, RWTH Aachen University, 52056 Aachen, Germany (e-mail: ohm@ient.rwth-aachen.de).

in some performance improvements. However, separate results for improvements stemming from interpolation and the affine motion model, respectively, suggest that the affine motion model does not really provide a significant advantage, particularly not in the way it was used for video compression by the authors.

Recently, a method [12] was introduced to deal specifically with zoom in video. Basically, it proposes to enhance H.264/AVC motion vector prediction such that it can deal better with zoom. However, the method still assumes equal motion vectors for entire blocks, an assumption that will be validated for a large set of sequences in Section II-B. However, it will be shown that this assumption does not hold true when the temporal distance between the reference frame and the frame to be encoded increases and a single motion vector per pixel becomes sensible. The idea from [12] has been used in the context of HEVC as well, see [13].

A promising path to advanced motion models has been taken in [14] and [15], work that will be described in more detail in Section II-B. This proposition has common ground with the work described here, that is, incorporating homographies into a video stream. While in this paper a dedicated algorithm computes a single homography as a representation of camera pan and/or zoom, in the cited work, a set of homographies is computed from the different point correspondences that are divided into clusters, each cluster represented by a homography. The method achieves good performance for high resolution videos containing strong zoom. The theoretical background therefore is provided in Section II-B. However it was not adopted into the HEVC standard.

B. *Contribution of this work*

We propose a method that allows to perform long-term global motion estimation using homographies. Unlike previous work ([14], [15]) our method allows to use a temporal distance between the frame to be coded and the warped frame. This provides an advantage since it allows to achieve gains on sequences in the common test set that were not reported previously, except for [16], which constitutes the basis of this paper. The theoretical foundations for the gains are provided here. In fact, the gains can be explained by the difference in scale between the frame to be encoded and the reference frame, and we experimentally derive an optimal scale ratio.

The rest of the paper is organized as follows. In Section II we provide a motivation for global motion compensation using homographies and show in what particular cases homographies constitute an advantage over block-based motion compensation in the context of HEVC. In Section III we present a simple distortion model that is then used to explain the observation that beyond sequences with repeated content, the presented method is also beneficial for sequences with change in scale, namely, zooming out. In Section IV we show the proposed enhancements to HM and briefly outline the methods used for homography computation and coding. Section V presents coding results for the system and we experimentally derive an optimal scale ratio. Finally we conclude with Section VI.

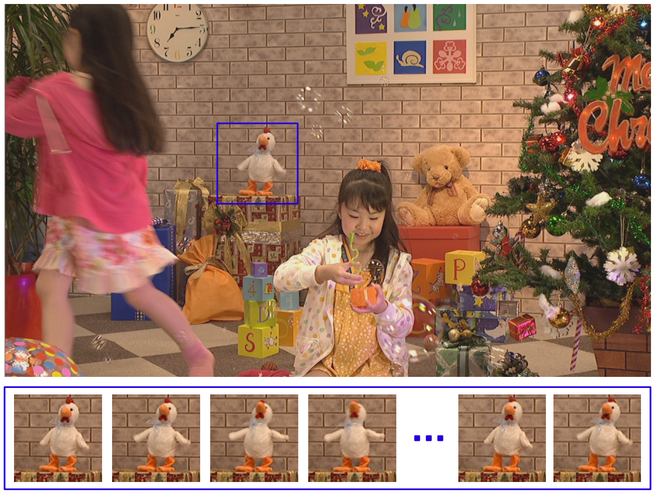


Fig. 1. PartyScene sequence contains some temporally periodic elements, see the box in the center. The cropped region at the bottom shows the plush toy is performing a quasi-periodic motion trajectory. The region has been cropped after compensating camera motion and zoom on the entire sequence.

II. Camera pan and zoom compensation

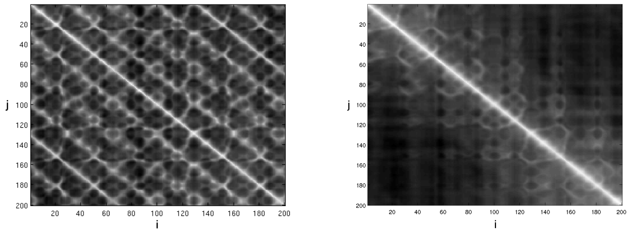
A. *Motivation*

Our initial motivation for employing global motion estimation was to be able to use methods similar to those presented in e.g. [17] with sequences containing camera motion and zoom. In fact, one of the main characteristics of dynamic or video textures is temporal repetition. As an example, in the PartyScene sequence, particularly for the cropped region from Figure 1, there is a temporally repetitive pattern. Note that the sequence contains camera motion and zoom, and effects arising from the latter have to be removed before an analysis of the scene itself can take place. Hence camera motion and zoom compensation was performed before cropping the region in Figure 1. It turns out that the region is temporally quasi-periodic, the period being 130 frames, see Figure 2 (a), where the lightest diagonal is 130 frames apart from the central diagonal. Furthermore, it appears that very similar frames are about 30 frames apart, a fact that could be exploited for compression purposes, as will be shown later. That camera motion and zoom compensation are crucial to recognize the periodicity in this part of the sequence is illustrated in Figure 2 (b), which shows the similarity matrix for the same region as in Figure 2 (a), however without camera motion and zoom compensation, that is, the original cropped region was taken.

It will be shown in this Section that, in the presence of zoom, homographies are necessary to perform long-term motion compensation between temporally distant frames. In fact, HEVC already incorporates motion compensation, but block-based motion compensation is not suitable for the problem described here.

B. *Deriving motion vectors from homographies*

Our method proposes to code homographies into the video stream for long-term compensation. To justify the necessity of homographies as an alternative to block-based motion compensation, motion vectors will be derived from homographies



(a) (b)

Fig. 2. Visualization of similarity matrices for the cropped region of the PartyScene sequence from the Figure 1. (a): after camera motion and zoom compensation. (b): original cropped region without compensation.

and then conceptually compared to block-based motion vectors as they are used in standard video coding. The basic equation for homographies states that for background objects, a relation between two pixel positions in two frames of a sequence can be described as follows:

$$\begin{pmatrix} x' \\ y' \\ n' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{1}$$

using homogeneous coordinates. Practically, this means that a pixel located at (x, y) will be located at position $(x'/n', y'/n')$ in the corresponding image. In this context a motion vector $\mathbf{v}$ would amount to:

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} x'/n' \\ y'/n' \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix}, \tag{2}$$

that is a vector pointing from the position (x, y) to $(x'/n', y'/n')$. A conceptual difference between motion vectors in video compression and view warping using homographies for the description of camera motion, is that every motion vector is valid for a group of pixels, that is a prediction unit in HEVC, while the motion vector $\mathbf{v}$ from equation (2) can differ from pixel to pixel. Generally speaking, the availability of a homography is only an advantage if motion vectors differ on a very fine scale.

C. *Visualization of homography-derived motion vectors*

One way of giving insight into how motion vectors spatially differ in practice, is by visualization. Hence, we want to visualize the spatial variation of motion vectors. It is clear that the motion vectors differ from pixel to pixel in most cases, especially in case of zoom. However differences below a certain threshold are not significant when it comes to compression. It is widely accepted that motion vector precision beyond $\frac{1}{8}$ of a pixel does not result in a better compression ratio in a realistic video encoder. This was already shown in [18], and is still widely accepted. Hence, to visualize on what scale significant motion vector differences occur, we will consider motion vector differences greater or equal to $\frac{1}{8}$ of a pixel, by visualizing them with arrows. We want to emphasize cases in which motion vector differences of more than $\frac{1}{8}$ of a pixel occur for spatially close pixels.

To visualize motion vector differences, the sequence BQSquare, which was used in the standardization process of HEVC, will serve as an example. The sequence has resolution WQVGA, and contains camera pan and zoom simultaneously, as can be seen in Figure 3.

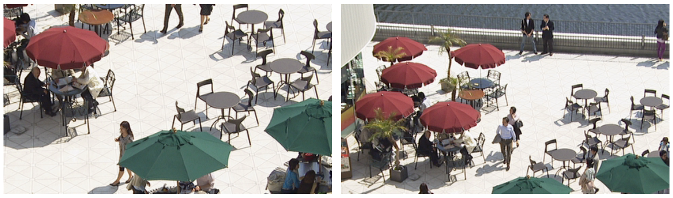


Fig. 3. Frames number 1 and 150 from BQSquare sequence with resolution WQVGA. The change in perspective comes from zooming out and camera pan.

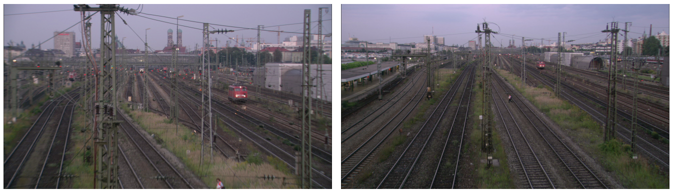


Fig. 4. Frames number 1 and 150 from the Station sequence. The change in perspective is extreme, considering that within a few seconds the scale of the scene has completely changed due to zooming out.

For the visualization of motion vector differences, only the upper left part of each frame in the image will be used. Figure 5 shows the cropped upper left region of the size 128x128 pixels of BQSquare. Arrows show the motion vector difference generated as follows: first, for every pixel in the image, a motion vector is derived. It is computed using a homography that describes the relation in between the first frame and the respective frame using the algorithm for camera motion compensation introduced in [16] and briefly summarized in Section IV-A. Second, the motion vector of the upper left pixel is taken as a reference and motion vector differences with respect to this vector are computed. In a final step quantization with a precision of $\frac{1}{8}$ of a pixel is applied. The arrows are normalized, that is, the length of the arrows is independent of the length of the motion vector difference. Hence only the change in direction can be illustrated. Every arrow stands for a region of 4x4 pixel, as visualizing a vector in the image for every pixel is not possible.

Since motion vector differences of less than $\frac{1}{8}$ of a pixel are not considered, the absence of arrows in large parts of the considered region of frame number 2 signifies that the motion vectors remain very similar in the considered area. In frame number 3, for the considered region of 128x128 pixels, four regions with a different motion vector arise. With increasing temporal distance, the differences between neighboring pixels increases which is illustrated in the second row of Figure 5 which shows the result for vector differences for frames 9, 14 and 19, respectively. Figure 6 is the same visualization, however for the sequence Station, which has 1080p resolution. Here the situation is different, that is, the motion vectors seem to differ spatially from pixel to pixel even when the considered frames are temporally close. This is due to the particularly strong zoom, see Figure 4, where the size of objects in the scene has completely changed during the time of 150 frames. This becomes particularly apparent when compared

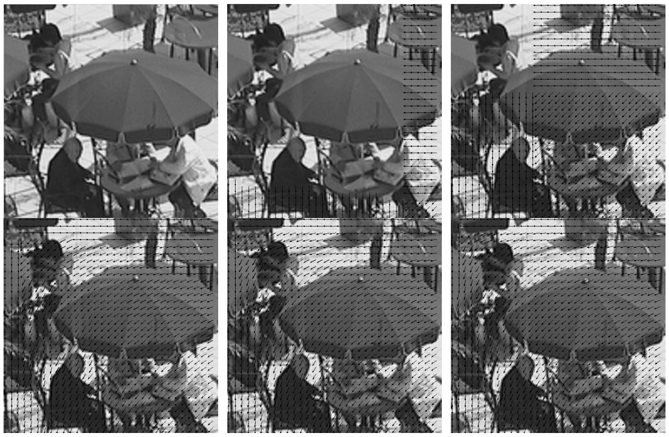


Fig. 5. Cropped upper left region of the size 128x128 pixels of BQSquare. Frames number 1, 2, 3, 9, 14, and 19 (left to right, top to bottom) are shown. In addition, arrows show the motion vector difference from the upper left motion vector and the motion vector at the current location. This illustrates that, in the presence of zoom, the difference between neighboring pixel motion vectors increases with the temporal distance of the considered frames.

to Figure 3, where the change in size is considerably less important. Hence it can be concluded that the spatial variation of motion vectors depends on the strength of the zoom in the sequence on one hand and the temporal distance between the considered frames on the other. While the visualization was useful to give a general impression of the spatial variation of motion vectors, we further want to provide a quantitative analysis in the sequel.

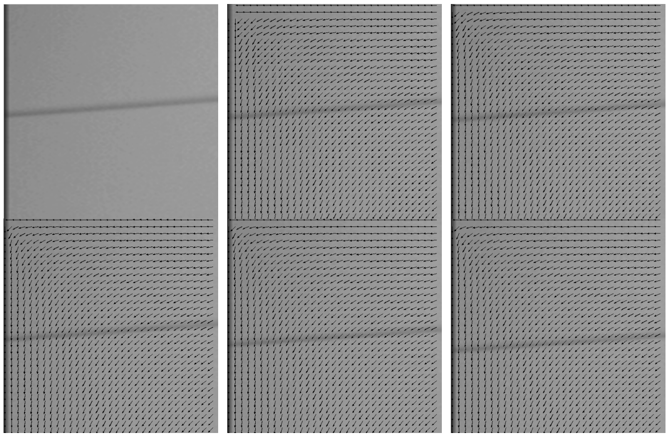


Fig. 6. Cropped upper left region of the size 128x128 pixels of Station. Frames number 1, 2, 3, 5, 7, and 9 (left to right, top to bottom) are shown. In addition, arrows show the motion vector difference from the upper left motion vector and the motion vector at the current location. All motion vectors are derived for the pixel position using a precomputed homography that can be assumed as being exact.

D. Lower and upper bounds for motion vector differences

As already mentioned background motion can be modeled using homographies. For convenience, we want to restate equations (1) and (2) in a less general way, namely by disregarding perspective effects. This means that h_{31} and h_{32} are zero, which will cause n' to be 1 which obviates it. Hence, for that case equation (1) can be written as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = A_h \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} h_{13} \\ h_{23} \end{pmatrix}, \tag{3}$$

with

$$A_h = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}. \tag{4}$$

This is in fact the affine motion model endowed with a displacement vector. The model is still able to represent zoom. Equation (2) for motion vectors still holds with $n' = 1$. Now let us define the two pixel locations $\mathbf{x}_0$ and $\mathbf{x}_1$. We are now interested in their respective motion vectors $\mathbf{v}_0$ and $\mathbf{v}_1$ and how they differ. Clearly, this depends on the distance between $\mathbf{x}_0$ and $\mathbf{x}_1$. Therefore we define the difference δ as

$$\delta = \mathbf{x}_1 - \mathbf{x}_0, \tag{5}$$

which makes $\|\delta\|_2$ the distance between $\mathbf{x}_0$ and $\mathbf{x}_1$. Then we have for the difference in corresponding vectors δ':

$$\begin{aligned} \delta' &= \mathbf{v}_1 - \mathbf{v}_0 \\ &= (\mathbf{x'}_1 - \mathbf{x}_1)(\mathbf{x'}_0 - \mathbf{x}_0) \\ &= A_h\mathbf{x}_1 - \mathbf{x}_1 - A_h\mathbf{x}_0 + \mathbf{x}_0 \\ &= (I - A_h)(\mathbf{x}_1 - \mathbf{x}_0) \\ &= (I - A_h)\delta, \end{aligned} \tag{6}$$

with I being the identity matrix. We are interested in some bounds on the difference in vectors. We can therefore consider the squared ℓ_2 norm of δ':

$$\begin{aligned} \|\delta'\|_2^2 &= \|\mathbf{v}_1 - \mathbf{v}_0\|_2^2 \\ &= \|(I - A_h)\delta\|_2^2 \\ &= ((I - A_h)\delta)^{\mathsf{T}}((I - A_h)\delta) \\ &= \delta^{\mathsf{T}}(I - A_h)^{\mathsf{T}}(I - A_h)\delta. \end{aligned} \tag{7}$$

The last line is a form of Rayleigh's Quotient if we set that $\|\delta\|_2^2 = 1$, that is fix $\mathbf{x}_0$ and $\mathbf{x}_1$ to be direct left or right, upper or lower neighbors. In fact we have that

$$\lambda_0 \geq \delta^{\mathsf{T}}(I - A_h)^{\mathsf{T}}(I - A_h)\delta \geq \lambda_1, \tag{8}$$

with λ_i being the eigenvalues of $(I - A_h)^{\mathsf{T}}(I - A_h)$. More generally, for any $\|\delta\|_2$, we have

$$\|\delta\|_2^2\lambda_0 \geq \delta^{\mathsf{T}}(I - A_h)^{\mathsf{T}}(I - A_h)\delta \geq \|\delta\|_2^2\lambda_1, \tag{9}$$

Considering that the eigenvalues λ_i are also the squares of the singular values of $(I - A_h)$, we have that

$$\|\delta\|_2 s_0 \geq \|\delta'\|_2 \geq \|\delta\|_2 s_1. \tag{10}$$

Hence using the homography we can find a lower bound for the motion vector differences between neighboring pixels.

In particular, we can take the sequence Tempete, the homography that describes the relation between the first frame and frame number 23 (it will be shown later that this is an optimal distance for long-term prediction for this sequence, and the zooming can be assumed constant throughout the sequence). Using equation (10) with the corresponding homography we can immediately compute a lower bound on how neighboring motion vectors will differ. From the singular values of $I - A_h$, it immediately follows that even for 2x2 blocks (i.e. $\|\delta\|_2^2 \leq 2$), $\|\delta'\|_2$ will not be small enough to assign all pixels in the block the same motion vector, that is, if we use a motion vector precision of $\frac{1}{8}$ of a pixel. In other words, it is sensible to use a different motion vector for every pixel, which can be achieved using homographies.

Similar results are obtained for the sequences including zoom, like Waterfall, Tempete, and Concrete, when considering high temporal distances (e.g. 15-35 frames), and for Station, where zoom is "stronger", this is already the case for close frames (distances of 1-4 frames).

Finally, two cases emerge where a homography may be sensible to describe motion compensation:

1) Particularly strong zoom in the video, as in the example of Station (1080p) sequence, see Figure 6.
2) Motion compensation with frames that are temporally distant in video containing zoom, see second row of Figure 5.

In the former case it has been shown that gains in video compression with HEVC can be achieved, see [15]. In this paper we will concentrate on the latter case. In general, when zooming is not so important, gains can be achieved by applying long-term motion compensation in the case of zooming out. While this Section has outlined that to perform long-term motion compensation in videos containing zoom the usage of homographies is necessary, the next Section will reveal why doing so can be advantageous in the first place. Finally, it will become clear that in both cases, the gains arise from the fact that we predict from images with finer scale. Understanding the real source of the gains that can be achieved with sequences containing zoom can allow to generalize gains for a very few sequences with very strong zoom to the more general case of zooming out.

III. Effects of texture scaling in motion compensation

A. Characterization of video distortion through compression

The power spectral density of a signal can be expressed in the following way:

$$\Phi_{ss}(\omega) = \mathrm{E}\left\{ \left| S(\omega)^2 \right| \right\}, \tag{11}$$

where $S(\omega)$ is the Discrete Time Fourier Transform of the sampled signal $s(n)$ and $\mathrm{E}\{\cdot\}$ denotes the expected value of a given expression. It is well known that that $R(D)$ (R denotes the rate, D the distortion) can be characterized for a discrete stationary Gaussian source, using a relation from [19]. In fact, it turns out that for such a signal with a spectral density $\Phi(\omega)$ and the distortion given in a parametric form by:

$$D_\theta = \frac{1}{2\pi} \int_{-\pi}^{\pi} \min\left[\theta, \Phi(\omega)\right] d\omega, \tag{12}$$

and

$$R(D_\theta) = \frac{1}{4\pi} \int_{-\pi}^{\pi} \max\left[0, \log\frac{\Phi(\omega)}{\theta}\right] d\omega. \tag{13}$$

The $R(D)$ curve is generated by assigning values from the interval $[0, \sup\, \Phi(\omega)]$ to θ. The notation $\sup(\cdot)$ is used for the supremum of a function.

More importantly, for the case described by equations (12) and (13), the distortion of the source can be characterized by the forward error channel, as shown in Figure 7. The source signal is filtered with a filter $H(\omega)$,

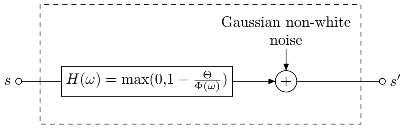


Fig. 7. The principle of the optimum forward error channel to model the distortion of a compressed source.

$$H(\omega) = \max\left[0, 1 - \frac{\theta}{\Phi(\omega)}\right] \tag{14}$$

and Gaussian noise with a spectral density

$$\Phi_{nn}(\omega) = \max\left[0, \theta(1 - \frac{\theta}{\Phi(\omega)})\right] \tag{15}$$

is added to the signal. This is an idealized description of how the compression will affect the signal in order to achieve a given rate and does not propose a real system that will have the given properties. Hence the proposed $R(D)$ curve is a simplification that may not be accurate. Furthermore, the assumption of a Gaussian distribution does not fit the reality with DCT coefficients, where the distribution for DCT coefficients of residual signals from hybrid video coding are closer to a Laplace distribution, see e.g. [20]. All the beforementioned notwithstanding, the given model provides useful insight in how the signal is affected by compression and will be very important to explain effects that can be observed with methods building upon HEVC. The rate distortion theory above applies to the 2-D case (i.e. images) in a similar fashion, but we relinquish to formulate the equations, since they have been presented in [21].

It is clear that the effect of compression on the images, which is the signal that is of interest, cannot be simply described by ideal low-pass filtering. However, if typical power spectral densities as they are assumed for images are used, the resulting effect of filtering with $H(\omega)$ can be described as non-ideal low-pass filtering. This result can be obtained when inserting, for instance, the power spectral density of an auto-regressive process, e.g. AR(1), into equation (14).

To complete this analysis we tried to verify the above in a practical example. The upper row of Figure 8 shows the first frame of the sequence Waterfall, which is mostly consisting of texture and therefore has desired signal characteristics.

To obtain an estimate of the power spectral density of an image, the relation given by equation (11), which defines the power spectral density as the expected value of the squared absolute value of the spectrum, is used as a basis. Practically, the expected value will be estimated by computing the mean over a series of square regions of the size 8×8 pixels. For the sake of simplicity the two chroma channels will be discarded and only luma will be used. The result as seen in the upper row of Figure 8 is computed as follows:

$$Z(e^{i\Omega}) = \frac{1}{N} \sum_{n=1}^{N} \left| S_n(e^{i\Omega}) \right|^2, \tag{16}$$

where Ω is a vector of spatial frequency, that is $\Omega = (\omega_x, \omega_y)$. Hence $Z(e^{i\Omega})$ is an estimate for the expected value of the power spectral density of the image. For convenience, the natural logarithm was plotted.

The above estimated power spectral density can be immediately used to verify qualitatively the assumed model for image degradation by compression. As an example, we used the first frame of the waterfall sequence. It was encoded with different QP's of 37 and 42. Both QP's are chosen such to have visible degradation, i.e. coding artifacts, which can be seen in the middle and lower row of Figure 8. The effect of low-pass filtering can be observed by considering the estimated power spectral densities that are provided along with the images. In fact, inserting the estimated power spectral density into equation (14) clearly makes $H(\omega)$ a low-pass filter which principally verifies the above for HEVC and the Waterfall sequence.

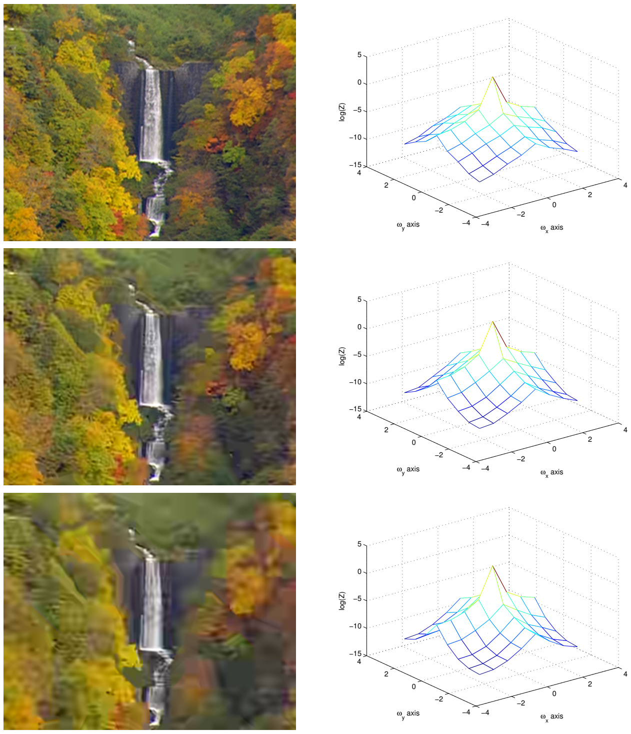


Fig. 8. The logarithm of the power spectral density is estimated for the original and two reconstructed images that were coded using the HM software at a QP of 37 and 42. In both cases the unaltered HM software was used. Differences can best be perceived when considering values close to $\{\omega_x, \omega_y\} = \{-\pi, -\pi\}$.

B. *Effects of changes in scale or perspective in video*

The notion of scale to be used here is very simple. It is assumed that in images that are seen in a video the sampling remains unchanged, which corresponds to the physical reality. Beyond the sampled signal (which is the image) it is assumed that there is a corresponding continuous signal, $s(t)$, which is sampled during the imaging process. While the sampling remains unchanged, zooming is modeled by simply *scaling* the continuous signal, that is the scaled signal is given by $s(\alpha t)$, where α is the *scaling factor* that will be used in the sequel, with α always being strictly positive.

Let us define a continuous 1-D signal $s(t)$ with its corresponding Fourier transform $\mathcal{F}(s(t)) = S(f)$. We are interested in the effect of scaling a signal (e.g. zooming in sequences), or more precisely in the effect of scaling on the Fourier transform. Indeed, the Fourier transform of a signal $s(\alpha t)$ is

$$S_\alpha(f) \quad = \quad \frac{1}{\alpha} S(\frac{f}{\alpha}), \tag{17}$$

for positive α.

Let us consider a spectrum $S(f)$ of to the signal $s(t)$ as in the example of Figure 9. The influence of scaling can be illustrated easily in this case, as a scaling with α in time implies a scaling with $\frac{1}{\alpha}$ in the frequency domain, and weighting with factor $\frac{1}{\alpha}$.

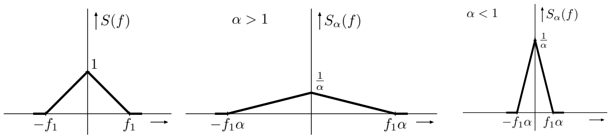


Fig. 9. Effect of scaling the signal $s(t)$ on the the Fourier transform $S(f)$.

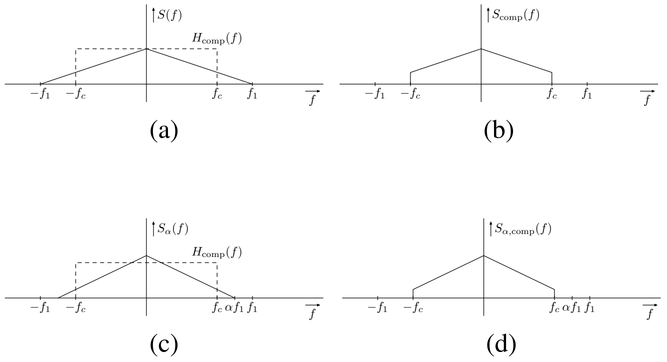
Practically this implies that when a signal is scaled with a factor $\alpha < 1$, that is, the shape of the signal broadens, the Fourier spectrum will become narrower and multiplied by $\frac{1}{\alpha} > 1$, which is an increase in amplitude. Speaking of video sequences, this property can easily be extended to the 2-D image case, where zooming out of a scene can be regarded as the scaling of the signal with an $\alpha > 1$ in both spatial directions. Considering that imaging is the process of mapping the real world on a 2-D plane, zooming in can be reasonably thought of as scaling in the real world. This analogy being made, we can now analyze the video compression process with special focus on zooming in videos.

It is clear that the imaging process entails low-pass filtering, since the physical pixels on CCD or CMOS sensors have a certain area and the light falling onto that area is spatially integrated (effects arising from optics put aside). Considering this, it is now interesting to review the process of scaling (zooming when speaking of video sequences). The signal falling on the sensor can be scaled as shown in Figure 9 to model zooming. The effect that details disappear in a video when zooming out can be modeled by saying that the details "are moving towards higher frequencies" and at some stage are filtered out somewhere in the image acquisition process. The idea itself may not be so important, but when combined with the effects occurring from compression, this can be used to explain the gains observed later in the results part of the paper.

C. *Prediction from compressed images with different scale*

To explain the gains with sequences containing change in scale, the assumption will be made that all the preceding notwithstanding, distortion from compression can be modeled with *ideal* low-pass filtering. Furthermore, we will assume that the image is a continuous 1-D signal with a corresponding Fourier Transform.

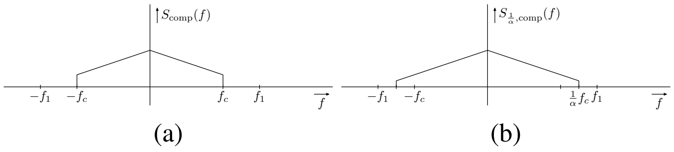
The compression process can hence be modeled as shown in Figure 10 (a). The Fourier Spectrum is simply low-pass filtered, so that the spectrum of the compressed image is given by $S_{\text{comp}}(f)$ from Figure 10 (b). In the presence of zooming out, the long-term picture buffer from Figure 12 contains images with a slightly finer scale. Using our model, we can assume these images to have a spectrum $S_\alpha(f)$, $\alpha < 1$, which is shown in (c). As all images are compressed with the same QP, and the scene does not change completely, we will assume the same $H_{\text{comp}}(f)$ for all images to model the low-pass filtering that results from the compression, see (c). The resulting Spectrum $S_{\alpha,\text{comp}}(f)$ is shown in (d). The difference between $S_{\text{comp}}(f)$ from (b) and $S_{\alpha,\text{comp}}(f)$ from (d) is that a part of the Spectrum is "saved" by the contraction of the Spectrum caused by scaling.



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

Fig. 10. (a) shows the Fourier Spectrum of an image being low-pass filtered, and (b) the resulting Spectrum of the compressed image given by $S_{\text{comp}}(f)$. The same image at a slightly finer scale has the Spectrum $S_\alpha(f)$ which is shown in (c). As this image was compressed with the same QP, the same $H_{\text{comp}}(f)$ can be used to model the low-pass filtering that results from the compression, see Figure 10 (c). The result is in turn the Spectrum $S_{\alpha,\text{comp}}(f)$ that is shown in Figure 10 (d). The difference with respect to $S_{\text{comp}}(f)$ from Figure 10 (b) is that a part of the Spectrum is "saved" by the contraction of the Spectrum caused by scaling.

Now, the two cases of prediction from images with equal and finer scale can be directly compared. The signal available in the former case is given in Figure 11 (a) which can be directly taken from Figure 10 (b). In the latter case, the compressed signal has to be scaled with factor $\frac{1}{\alpha}$ first in order to match the current image (in practice this is done by warping, see Figure 12). The process results in what is given in Figure 11 (b), that is, $S_{\frac{1}{\alpha},\text{comp}}(f)$, which is obtained by re-scaling $S_{\alpha,\text{comp}}(f)$ from Figure 10 (d). Comparing (a) and (b) shows that in (b) more "high-frequent" information is available, that could be useful since the original $S(f)$ should be predicted.



(a) (b)

Fig. 11. (a) is the Spectrum already shown in Figure 10 (b), which is now compared to the re-scaled signal $S_{\frac{1}{\alpha},\text{comp}}(f)$. The latter is obtained by scaling the $S_{\alpha,\text{comp}}(f)$ with the factor $\frac{1}{\alpha}$. It becomes apparent that in case (b) more "high frequent" information is available, and since ideally the full signal as seen in Figure 10 (a) should be reconstructed, using $S_{\alpha,\text{comp}}(f)$ seems to pose an advantage.

IV. Proposed method

We have shown that homographies are useful to achieve a motion vector precision of $\frac{1}{8}$ of a pixel in every pixel location in the presence of zoom and long-term motion estimation. Furthermore, we have shown that it can be assumed that details can be better preserved with long-term motion compensation from compressed images, that is, prediction from finer scale distorted images is advantageous. The two taken together, we propose a system as presented in Figure 12. It uses an extra long-term picture buffer where pictures are stored and then a picture with a given distance is warped into the perspective of the frame to be encoded. The latter is then inserted into the reference picture buffer for the purpose of motion compensation. We want to emphasize that the enhanced encoder and decoder use the same number of reference pictures, that is, the oldest reference picture is simply replaced by the warped long-term reference.

To allow the decoder to perform the same procedure the homography is coded into the bitstream. Instead of coding the floating point values of the homography themselves we encoded motion vectors generated from the homography acting on the corner points of the image. This allows the decoder to derive the homography. For the decoder it is sufficient to have four motion vectors to derive one homography per frame and thereby perform the same warping operation at the decoder. In practice, the bits stemming from those four motion vectors are negligible with respect to the overall bitrate, even for high QP values.

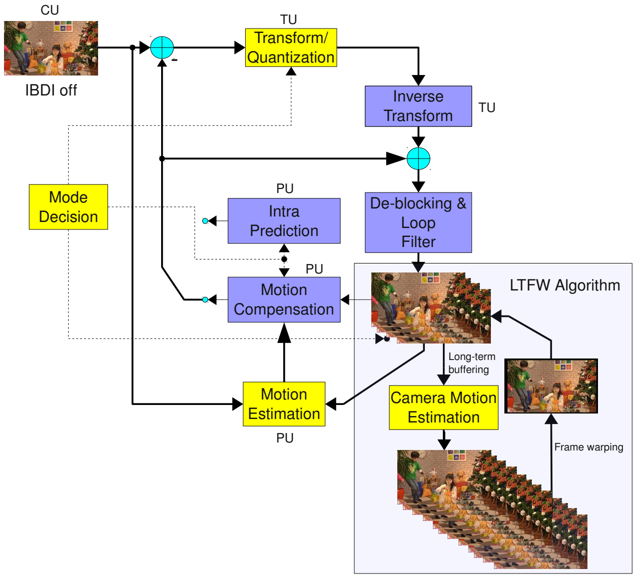


Fig. 12. Overall concept of the proposed method. We use long-term frame warping (LTFW) in a separate frame buffer to warp a temporally distant frame into the perspective of the currently coded frame. The generated frame is then inserted into the reference frame buffer and can be used for motion compensation.

A. Homography estimation

The assumption behind the techniques presented here is that global motion can be modeled with homographies. However it is not a trivial problem to compute homographies between any two frames of a sequence. The algorithm we use in this paper has been presented in [16]. We want to only outline the most important elements of the method:

1) Points that are well-suited for tracking are selected and tracked over time using the Lukas-Kanade algorithm.
2) A robust method is employed to derive homographies from point correspondences.
3) An algorithm automatically decides when to restart the estimation, i.e. when a new reference frame is selected to re-initialize the tracking from. Matrix multiplication of the homographies is used to relate frames after the re-initialization.

Another important factor in the warping process is the interpolation method. In a similar fashion as in [11], we use bicubic spline interpolation which has proven to perform similarly as the interpolation method used in HEVC and hence constitutes a state-of-the-art interpolation method for video compression.

B. Complexity

The proposed scheme adds to both, the overall compuational and space complexity. The global motion estimation with homographies as described in section IV-A consists mainly in tracking a number of points using the Lukas-Kanade algorithm. In our particular implementation a real-time motion estimation with sequences of resolution up to CIF is possible. For the results presented in this paper, and the sequences with low resolution (up to CIF), global motion estimation was estimated in real time, that is about 30 frames a second. If the number of tracked points is limited, this is also possible with higher resolution video, however, for the results presented here, the processing frequency for a 720p sequence was of about 15 frames per second. We shall mention that this does not really add to the overall encoder complexity, since with the HM software that we used for our experiements, only about 2 frames can be encoded per minute on the same machine. The additional decoder complexity is very limited, since the additional operations necessary for decoding our streams is limited to warping one picture in the decoded picture buffer and hence decoding time is comparable to HM.

As for the space complexity, the basic scheme as described in Figure 12 requires a number of additional frames to be stored. In particular, with a frame distance of 20 between the frame to be coded and the warped frame, 16 additional frames need to be kept in memory. However, we want to stress that those are not additional reference frames. In addition, a fixed frame distance is only important for our later experiments in this paper, and is not required in general. By relaxation of this condition, one could use varying long-term frame distances, and thereby considerably decrease the number of stored frames. It will be shown later (in Figure 15) that varying the distance within certain bounds does not have a strong effect on the compression ratio.

V. Results

The results shown here compare the conventional HM2.2 with HM2.2 enhanced by homography-based long-term prediction. In fact we use the same algorithm as we did in [16], however with the high efficiency setting instead of low complexity. Also a broader range of sequences is used for the experiments.

A. Results with fixed frame distance

Tests of the performance of the algorithm presented in Figure 12, that is, a system that warps old reference frames into the perspective of the frame to be encoded are presented here. A fixed temporal distance of 20 frames between encoded and warped reference frame is used. A more detailed analysis on how the temporal distance affects the gains achieved with the method will be conducted in Section V-C.

The encoder settings are listed in Table I and are identical to those listed in [22] that were used for the core experiments in JCTVC after the CfP for HEVC. We use the low delay/high efficiency setting, up to the RateGOPSize that has been changed from 4 to 1. The parameters of this setting are summarized in Table I.

Parameter	Value
GOP Structure	IBBB
QP I	22, 27, 32, 37
QP B	23, 28, 33, 38
Frame Rate	24, 30, 50 or 60 Frames/s
Number Reference Frames	4
Search Range	64 Pixels
Maximum Coding Unit Width	64 Pixels
Maximum Coding Unit Height	64 Pixels
Maximum Coding Partition Depth	4
Maximum TU Transform Size	2^5
Minimum TU Transform Size	2^2
Inter TU Maximum Depth	3
Intra TU Maximum Depth	3
Entropy Coding Mode	CABAC
Hierarchical B Coding	Off
Deblocking Loop Filter	On
Merge Mode	On
Adaptive Loop Filter	On

TABLE I
HM2.2 Encoder setup for high efficiency/low delay.

The coding results are given in Table II. The method presented in Figure 12 seems to introduce a lot of novel aspects to HEVC based compression since it incorporates elements that used to play a role in outdated video coding standards under new conditions, that is, state-of-the-art global motion estimation, transmission and interpolation. As already shown in Figure 1, temporal repetition accounts for some of the gains. However looking closer at the gains another characteristic is typical for the sequences where gains are observed. In fact, unlike PartyScene, where there is zooming in, in Waterfall, Tempete, Concrete Jets, Station and BQSquare there is zooming out. It is now important to note that zooming out in the scene together with a setting as shown in Figure 12 implies prediction from finer scale images, the case described in Section III-C.

All the sequences presented have a frame rate of 30 frames per second, except for PartyScene with 50 frames per second and BQSquare with 60. The resolution is given in Table II, and furthermore the first 300 frames were used for the BQSquare sequence, the first 150 for Station and the first 200 for Jets.

For comparison, with a QP setting of GOP related QP of 4, that is varying QP from frame to frame, for the reference encoder and enhanced system a Bjøntegaard Delta Rate of -27.5 % can be achieved for the Waterfall sequence and -7.5 % for the BQSquare sequence. Hence the coding gains are slightly lower, but for the following discussion on optimal distances between reference frame and frame to be encoded, the usage of equal QP is a natural choice in order to get clear results that are not affected by changes in QP from frame to frame.

It will be seen in Section V-C that the gains depend on the scale ratio between the warped frame and the frame to be encoded. Slightly better results than those presented here can be achieved by varying this ratio, but the distance $\Delta = 20$ is, in most cases, close to the optimum, see Figure 15. The only exception is the Station sequence, were the zooming out is very strong as pointed out previously. Hence we can mention the results for a frame distance of $\Delta = 4$ for Station, which is a Δ-Rate of 38.141%, which is significantly better than the 15.112% that are achieved with $\Delta = 20$.

As mentioned before, gains for the Station sequence were already presented in [14], [15], but no details on the implementation nor the precise realization are given. We want to emphasize that the sequence BQSquare (for which we could achieve gains) was included in the common conditions of the core experiments, see [22], and therefore tested in [14], [15], but there no gains could be acheived for BQSquare or any other of the test sequences included in the test set, since the mechanism presented here was not discovered at the time.

Sequence	Res.	ΔPSNR	ΔRate
Waterfall	336x272	1.393 dB	-28.765 %
BQSquare	WQVGA	0.342 dB	-7.632 %
Concrete	CIF	0.586 dB	-10.585 %
Tempete	CIF	0.727 dB	-14.949 %
PartyScene	WQVGA	0.222 dB	-4.576 %
Sheriff	360p	0.234 dB	-4.813 %
Jets	720p	0.634 dB	-26.811 %
Station	1080p	0.329 dB	-15.112 %

TABLE II
BJONTEGAARD DELTA RESULTS FOR HM2.2 AND A FIXED TEMPORAL DISTANCE OF $\Delta = 20$ FRAMES BETWEEN REFERENCE FRAME AND FRAME TO BE ENCODED.

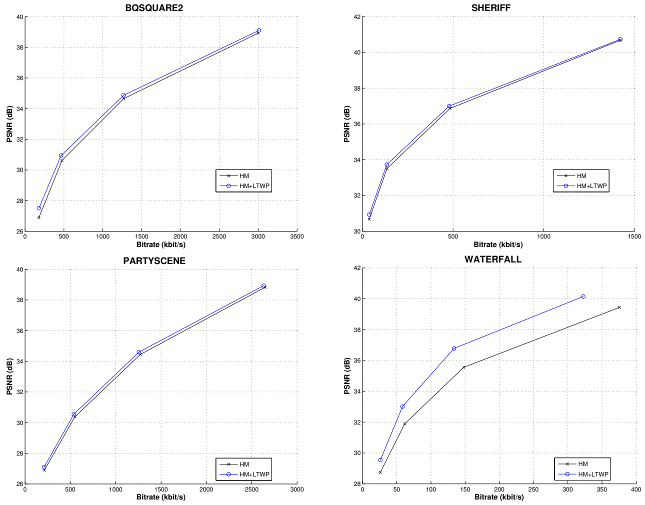
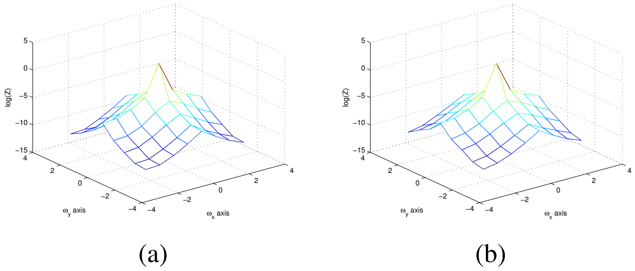


Fig. 13. Rate-distortion curves for a set of sequences and with a temporal distance of 20 frames between encoded and warped reference frame.

B. Power spectral density of reconstructed frames

To verify the above model on the gains achieved, we considered estimating the power spectral density of reconstructed frames from both the original encoder and the system using LTFW as described in this paper. The power spectral density of an image can be estimated as described previously by using equation (16). For the case of the Waterfall sequence and QP 37 the power spectral density of the last frame was estimated for both the encoding method with the long-term warped prediction and the original encoder and shown in Figure 14 where (a) is the original encoder and (b) the enhanced system. The gains in terms of PSNR that are of about 1 dB for the case of QP=37 (see Figure 13). This shows that the conjecture that by using the presented long-term warping technique, details contained in "higher frequency" components of the signal can be better predicted. The effect does not appear in an idealized fashion as presented in Figure 11, but the model seems to capture well the principal mechanism behind the presented gains for sequences with zooming out.



(a)

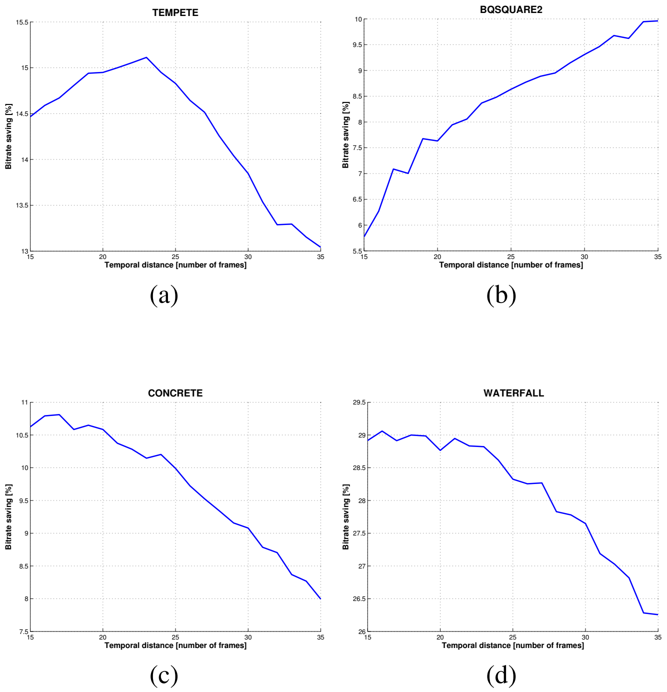
(b)

Fig. 14. The estimated power spectral density for the last frame of the sequence Waterfall. (a) is the original encoder and (b) the enhanced system with warping. It shows that the gains in PSNR are reflected in the fact that the power spectral density contains more "high frequency" components when warping is used. Differences can only be perceived when considering values close to $\{\omega_x, \omega_y\} = \{-\pi, -\pi\}$.

C. Results with different frame distances

The usage of homographies in video compression allows the usage of far distant frames for reference despite camera motion and zoom. With the principle of prediction from images with different scale, we will now try to find an optimal relative scale for prediction. Assuming that the focal length f in the frame to be encoded is given with $f = 1$, we will denote the focal length of the warped framed as f_Δ, with Δ being the temporal distance measured in number of frames. This ensures that the ratio $\frac{f_\Delta}{f}$ is equal to f_Δ.

We encoded the entire sequences multiple times using a range for Δ of 15 to 35. The optimal Δ for each sequence is given in Table III. For the sake of conciseness, we will not provide any details how we estimated a constant focal length ratio over the whole sequence but it will suffice to say that for the sequences that we analyzed the ratio remained almost constant throughout the sequence and we used regression to get an estimate for the latter. Hence Table III also provides an estimate for the optimal value of this ratio.



(a)

(b)

(c)

(d)

Fig. 15. Bjøntegaard Delta Rate savings for a set of sequences containing camera motion and zooming out. Varying the distance between the frame to be coded and the reference frame causes change to the compression ratio. The given examples show that the variation of the performance around the optimal frame distance is not too strong.

For the purpose the Bjøntegaard Delta Rate savings depending on the temporal distance between the frame to be coded and the reference frame that is warped are presented in Figure 15. This Figure illustrates that a fixed temporal distance may not be necessary since variation around the optimum is oftentimes not so strong, which would allow for flexible frame distance requiring less memory in the encoder and decoder.

Sequence	optimal $\boldsymbol{\Delta}$	f_Δ
Waterfall	21	0.9566
BQSquare	> 35	< 0.9536
Concrete	17	0.9463
Tempete	23	0.91

TABLE III

Table contains values for optimal Δ between warped reference frame and frame to be coded. Furthermore the estimated relation between focal lengths is computed for the respective value of Δ.

From the values presented in Table III, it can be concluded that the optimal relative focal length is between 0.91 and 0.95 if Bjøntegaard Delta is considered (that is, the overall optimal Δ is computed taking all QP's into consideration at the same time), regardless of the spatial characteristics of the encoded images.

A first application of the found ratio was already provided in Section V-A, where the gains for Station were shown not only for the long-term, but also for warped frames within the reference buffer, that is, $\Delta = 4$. Indeed, the relative focal length in that case is 0.95, compared to approximately 0.8 in the case of $\Delta = 20$, where the rate savings achieved are less than half, see Table II.

The analysis in Section III-C may suggest the following: the smaller the ratio f_Δ/f, the stronger the effect of acquiring high frequency information. However, two limiting factors have to be mentioned. First, when the scale ratio is low this means that not the entire image can be predicted but only the portion that is seen in the fine scale view. Second, when the scale ratio becomes very small, the warping process turns into decimation (or subsampling) which may incur aliasing. Analyzing this problem is a future research field and could lead to an interpolation method that incorporates low-pass filtering. However this analysis is beyond the scope of this paper.

VI. Conclusion and Future Work

We have shown that in the context of HEVC, global motion compensation using homographies is sensible in the special cases of repeated content and/or zoom. Our system can compute camera motion and zoom throughout an entire sequence and in this paper we have shown how this can be used to improve the compression efficiency in a video coding system that is based on HEVC. The results presented here are specific for HEVC and our method. In future, we plan to change the interpolation method (e.g. one incorporating low-pass filtering) or enhancing the system to allow for multiple homographies per frame. The latter would render the tracking a more challenging task since the long-term tracking of objects in a video is still considered a challenge in computer vision. Furthermore, to take advantage of the presented effect, the objects would have to undergo change in scale, a circumstance that renders the tracking particularly challenging. All considered, this constitutes a future option in video coding that could help improve compression beyond the current state-of-the-art.

References

[1] A. Smolic, T. Sikora, and J.-R. Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1227 –1242, Dec. 1999.

[2] F. Dufaux and F. Moscheni, "Background mosaicking for low bit rate video coding," in *IEEE International Conference on Image Processing*, Sep. 1996, vol. 1, pp. 673 –676.

[3] T. Wiegand, E. Steinbach, and B. Girod, "Long-term memory prediction using affine motion compensation," in *IEEE International Conference on Image Processing*, 1999, vol. 1, pp. 51 –55.

[4] T. Wiegand, Xiaozheng Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 70 –84, Feb. 1999.

[5] P. Ndjiki-Nya, T. Hinz, A. Smolic, and T. Wiegand, "A generic and automatic content-based approach for improved H.264/MPEG4-AVC video coding," in *Proceedings of the IEEE International Conference on Image Processing*, Sep. 2005, vol. 2, pp. 874–877.

[6] S. Ierodiaconou, J. Byrne, D. R. Bull, D. Redmill, and P. Hill, "Unsupervised Image Compression using Graphcut Texture Synthesis," in *Proceedings of the IEEE International Conference on Image Processing*, Nov. 2009.

[7] M. Bosch, Fengqing Zhu, and E.J. Delp, "Segmentation-based video compression using texture and motion models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1366 –1377, Nov. 2011.

[8] Aljoscha Smolic, Yuri Vatis, Heiko Schwarz, and Thomas Wiegand, "Improved H.264/AVC coding using long-term global motion compensation," in *Proc. VCIP 2004, SPIE Visual Communications & Image Processing*, Jan. 2004.

[9] D. Springer, F. Simmet, D. Niederkorn, and A. Kaup, "Compression of 2D navigation views with rotational and translational motion," in *SPIE Electronic Imaging - Visual Information Processing and Communication III*, San Francisco, CA, USA, January 2012, vol. 8305.

[10] D. Springer, Franz Simmet, Dieter Niederkorn, and A. Kaup, "Compression of 2D and 3D navigation video sequences using skip mode masking of static areas," in *Picture Coding Symposium*, Krakow, Poland, May 2012.

[11] H. Lakshman, H. Schwarz, T. Blu, and T. Wiegand, "Generalized interpolation for motion compensated prediction," in *Proceedings of the IEEE International Conference on Image Processing*, Sep. 2011, pp. 1213 –1216.

[12] H. Yuan, Y. Chang, Z. Lu, and Y. Ma, "Model based motion vector predictor for zoom motion," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 787 –790, Sep. 2010.

[13] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Parametric motion vector prediction for hybrid video coding," in *Picture Coding Symposium*, Krakow, Poland, May 2012, pp. 381–384.

[14] S. Park, J. Sung, J. Young Park, and B.-M. Jeon, "TE3: Motion compensation with adaptive warped reference," JCTVC-B022, 2nd Meeting, Geneva, Switzerland, 2010.

[15] S. Park, J. Park, and B.-M. Jeon, "TE3: Motion compensation with adaptive warped reference," JCTVC-C033, 3rd Meeting, Guangzhou, China, 2010.

[16] A. Stojanovic, C. Müller, and M. Knorr, "Improved temporal texture coding using camera motion estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012.

[17] Aleksandar Stojanovic and Philipp Kosse, "Extended dynamic texture prediction for H.264/AVC inter coding," in *Proceedings of the IEEE International Conference on Image Processing*, Hong Kong, People's Republic of China, Nov. 2010, pp. 2045 – 2048.

[18] Bernd Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604 –612, Apr. 1993.

[19] Toby Berger, *Rate Distortion Theory*, Prentice Hall, Englewood Cliffs, N.J., 1971.

[20] E.Y. Lam and J.W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1661 –1666, oct 2000.

[21] Bernd Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140 –1154, Aug. 1987.

[22] Frank Bossen, "Common test conditions and software reference configuration," JCTVC-D600, 4th Meeting, Daegu, Korea, Jan. 2011.

Aleksandar Stojanovic received his diploma degree in Electrical Engineering and Information Technology in 2007 and his PhD degree in 2013 both from RWTH Aachen University. Currently he is a postdoctoral research associate at the Center for Security, Reliability and Trust (SnT) in Luxembourg.

Jens-Rainer Ohm (M '92) received the Dipl.-Ing. degree in 1985, the Dr.-Ing. degree in 1990, and the habil. degree in 1997, all from Technical University of Berlin (TUB), Germany. From 1985 to 1995, he was a research associate with the Institute of Telecommunication at TUB. Between 1992 and 2000, he has also served as lecturer on topics of digital image processing, coding and transmission at TUB. From 1996 to 2000, he was project coordinator at Heinrich Hertz Institute (HHI) in Berlin. In 2000, he was appointed full professor and since then he holds the chair position of the Institute of Communication Engineering at RWTH Aachen University, Germany. His research and teaching activities cover the areas of motion-compensated, stereoscopic and 3-D image processing, multimedia signal coding, transmission and content description, audio signal analysis, as well as fundamental topics of signal processing and digital communication systems. Since 1998, he participates in the work of the Moving Picture Experts Group (MPEG). He has been chairing/co-chairing various standardization activities in video coding, namely the MPEG Video Subgroup since 2002, the Joint Video Team (JVT) of MPEG and ITU-T SG 16 VCEG from 2005 to 2009, and currently, the Joint Collaborative Team on Video Coding (JCT-VC) as well as the Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V). Prof. Ohm has authored textbooks on multimedia signal processing, analysis and coding, on communication engineering and signal transmission, as well as numerous papers in the fields mentioned above. He is member of various professional organizations including IEEE, VDE/ITG, EURASIP and AES.