EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Perspective transform motion modeling for improved side information creation

Pedro Monteiro[1*], João Ascenso[2] and Fernando Pereira[1]

Abstract

The distributed video coding (DVC) paradigm is based on two well-known information theory results: the Slepian-Wolf and Wyner-Ziv theorems. In a DVC codec, the video signal correlation is mostly exploited at the decoder, providing a flexible distribution of the computational complexity between the encoder and the decoder and error robustness to channel errors. To exploit the temporal correlation, an estimate of the original frame to code, well-known as side information, is typically created at the decoder. One popular approach to side information creation is to perform frame interpolation using a translational motion model derived from already decoded frames. However, this translational model fails to estimate complex camera motions, such as zooms and rotations, and is not accurate enough to estimate the true trajectories of scene objects. In this paper, a new side information creation framework integrating perspective transform motion modeling is proposed. This solution is able to better locally track the trajectories and deformations of each object and increase the accuracy of the overall side information estimation process. Experimental results show peak signal-to-noise ratio gains of up to 1 dB in side information quality and up to 0.5 dB in rate-distortion performance for some video sequences regarding state-of-the-art alternative solutions.

Keywords: Distributed video coding; Side information; Perspective motion model; Frame interpolation

1. Introduction

Nowadays, image, video, and audio digital coding technologies are widely used by a significant amount of the world population. This leads to a huge volume of data being transmitted and stored, especially when video information is involved. The key objective of digital audiovisual coding techniques is to compress the original information into the minimum number of bits for a target decoded signal quality, eventually also fulfilling other relevant requirements such as error resilience, random access, and scalability. Nowadays, most digital video-enabled services and devices use the popular H.264/AVC (Advanced Video Coding) standard, a joint ITU-T and ISO/IEC MPEG effort, which typically provides up to 50% compression efficiency gain (this means about half the rate for the same perceptual quality) compared to the previously available standards [1]. However, the state of the art on predictive video coding has been evolving, and in January 2013, another milestone on predictive video coding has emerged, the so-called High Efficiency Video Coding (HEVC) standard [2], which brings again about 50% additional compression compared to the H.264/AVC High profile solution [3] while increasing the encoding complexity. On one hand, the new upcoming HEVC standard has a higher complexity encoder, several times more complex than H.264/AVC encoders, and a real-time implementation will be a subject of research in the near future [4]. On the other hand, an HEVC decoder has a rather similar complexity to an H.264/AVC decoder, which means that the standardization trend of developing rather high encoder complexity when compared to the decoder complexity is still present. This type of complexity budget suits well the down-link broadcast model, where few powerful encoders provide coded content to many simpler and cheaper decoders. However, some emerging video applications are not well characterized by the down-link model but rather follow an up-link model, where some simple devices deliver information to a central, eventually rather complex, receiver. Examples of these applications are wireless low-power video surveillance, visual sensor networks, mobile video communications, and deep-space applications. Typically, these

* Correspondence: pedro.monteiro@lx.it.pt
[1]Instituto Superior Técnico – Instituto de Telecomunicações, Lisbon 1049-001, Portugal
Full list of author information is available at the end of the article

emerging applications require light encoding or at least a flexible distribution of the video codec complexity, robustness to packet losses, high compression efficiency, and, often, low latency/delay as well. These novel requirements and needs led to the emergence of a new video coding paradigm based on the Slepian-Wolf and Wyner-Ziv (WZ) Information Theory theorems [5,6], well known as *distributed video coding* (DVC). DVC targets light video encoding systems while theoretically achieving the same compression efficiency as the best predictive video coding schemes available (for specific conditions defined by the theorems), improved error resilience, and codec-independent scalability [7]. Moreover, DVC allows exploiting the inter-view correlation in multiview video scenarios with the architectural advantage that the encoders do not need to communicate among them. In the DVC paradigm, it is crucial that the correlation between the video frames can be efficiently exploited at the decoder side (as the encoder is not anymore exploiting this correlation as in predictive video coding). Therefore, the decoder module responsible for this task, the *side information (SI) creation module*, is rather critical as its performance strongly impacts the overall DVC codec performance. Typically, to create each side information frame, a translational motion model, representing the motion of each block with a motion vector, is used in most DVC codecs [8]. However, this motion representation is not powerful and accurate enough to efficiently estimate complex motions, like rotations, zooms, and object deformations, and may lead to motion field discontinuities and inaccuracies. To overcome the translational motion model limitations, it is necessary to exploit the capabilities of more advanced motion models to estimate the SI frame. In addition, the SI creation techniques can also be used to enhance the performance of a predictive video decoder when errors corrupt the video bitstream; for example, when an entire video frame is lost (a plausible occurrence in packet loss networks), SI creation techniques can conceal this frame rather efficiently. Other possible use is frame rate up-conversion, notably when the encoder drops some frames to save bitrate to meet the constraints of a bandwidth-limited channel; in this case, the decoder can still obtain a reliable estimate of the lost frame, minimizing the error propagation in a predictive group of pictures (GOP) structure.

In this context, this paper proposes a novel side information creation framework that exploits a perspective transform motion model to more accurately represent the temporal correlation between the video frames, thus obtaining better SI quality when compared with the typical translation motion model SI creation solutions. The proposed SI creation solution obtains two estimations for each SI block: one using the popular translational motion-compensated frame interpolation (MCFI) method [9] and another using the proposed perspective transform motion model; then, for each block, the best approach is selected and the final SI frame is created by appropriately combining, at the block level, the two preliminary SI estimations. In addition, to efficiently deal with occlusions, motion models are created for both temporal directions between reference frames (this means backward to forward and forward to backward) and the results are appropriately fused.

The rest of this paper is organized as follows: in Section 2, some relevant SI creation techniques and DVC solutions using advanced motion models available in the literature are reviewed; next, Section 3 presents the architecture and walkthrough of the proposed SI creation solution while Section 4 makes a detailed description of the proposed techniques; in Section 5, the performance of the proposed SI creation solution is assessed in terms of both SI quality through a peak signal-to-noise ratio (PSNR) metric and rate-distortion (RD) performance in the context of a state-of-the-art DVC solution; and finally, Section 6 presents some conclusions and future work.

2. Reviewing side information creation techniques and advanced motion models

The early Stanford DVC solution is characterized by frame-based Slepian-Wolf coding, at the beginning using turbo codes and later low-density parity-check (LDPC) codes, and assumes the availability of a feedback channel to perform rate control at the decoder [10]. The DISCOVER codec adopts the same architecture and includes state-of-the-art techniques for most of the techniques, such as LDPC codes, advanced (translational) motion-compensated frame interpolation, online estimation of the correlation noise model, and a cyclic redundancy check (CRC) error detection code [8]. Although all DVC decoder techniques are important to reach the best RD performance, SI creation has a critical impact on the distributed video codec compression efficiency. In the past, several SI creation techniques only using past decoded frames to create the SI have been proposed. In [9], a translational SI creation framework using a regularization criterion for motion estimation, adaptive search range, two block sizes, and a spatial motion vector median filter is proposed. In [11], mesh-based motion estimation and interpolation aims to better represent the motion field, especially for scenes composed by large objects and/or scenes with dominant camera motion. In [12], the motion between the K previously decoded frames $(i - K,..., i - 1)$ is tracked to extrapolate the motion field for frame i using a Kalman filtering approach. In [13], a method called high-order motion interpolation (HOMI) was proposed where the SI is created with two reference frames from the past and two reference frames from the future. The motion trajectory is interpolated using information obtained from four frames instead of the (typical) two-frame scenario as in the MCFI method, thus allowing the adoption of more complex motion models, such as non-linear motion models. Bjontegaard delta PSNR (BD-PSNR) improvements of 0.044 to 0.141 dB are achieved when compared to the usual DISCOVER MCFI approach. In [14], an autoregressive (AR) model is used to generate the SI frames targeting a low-delay DVC scenario. Each SI pixel is calculated as a linear weighted summation of pixels within a window in the previous reconstructed frames. An accurate AR model must be estimated to obtain high-quality SI; in such case, two weighting coefficient sets are computed for each SI block, which allow creating two SI frames which are fused together with a simple extrapolation SI frame according to a probability model. The RD performance results are rather promising when compared to other state-of-the-art motion extrapolation results available in the literature. In [15], a SI creation approach is proposed where the first steps include forward and bidirectional YUV motion estimation algorithms with variable block size. Then, spatial motion smoothing and motion refinement is performed and an adaptive overlapped block motion compensation algorithm is applied to a few selected neighboring motion vectors. The results show up to 0.4-dB improvements when compared to the DISCOVER MCFI-based method. In [16], an optical flow-based frame interpolation method was proposed and combined with an overlapped block motion compensation (OBMC)-based frame interpolation method. A multihypothesis transform-domain DVC decoder exploits the SI frames interpolated by both schemes with the help of a weighted joint distribution obtained from the (individual) correlation noise distribution associated to each SI frame. The total variation-L1 (TV-L1) norm optical flow algorithm was used to compute two flow fields, the backward and the forward flow, leading to two SI frame estimations which are then combined. This type of approach was also followed in [17] where a dense motion field was computed with pixel-recursive Cafforio-Rocca algorithm adapted to the frame interpolation context. The RD performance improvements are up to 2% gains over the DISCOVER MCFI scheme. In [18], it is proposed to combine global and local (MCFI-based) motion compensation estimation at the decoder side to improve the SI quality. To create a global motion-compensated (GMC) estimation, Scale-invariant feature transform (SIFT) features are extracted and a global affine motion model is computed at the encoder side. Then, the global affine motion model parameters are transmitted to the decoder to generate the GMC estimation, which is fused with the MCFI estimation, at the block level, to obtain the final SI frame. With the proposed SI creation method, it is possible to systematically outperform the standard H.264/AVC Intra and H.264/AVC zero motion coding solutions for all the video sequences evaluated. However, despite the RD performance improvements shown, this approach increases the encoding complexity significantly in the calculation of the SIFT features, matching, and global motion modeling estimation, which is not desirable in a DVC scenario where low encoder complexity is targeted. Thus, it has become clear that to further improve the SI quality produced by motion-compensated frame interpolation schemes (thus obtaining further compression efficiency gains), more accurate and reliable motion models must be used. A promising approach is to estimate the deformation of objects along time with higher-order motion models that are able to more accurately describe the geometric transformations between reference frames when compared to the simpler translational motion models. With more advanced motion models, it is possible to obtain better predictions in predicted video coding schemes (and lower bitrates for the residual signals) and higher-quality SI frames in distributed video coding scheme (and fewer SI errors to correct with lower bitrates). These motion models allow warping (transforming) a quadrilateral of any size and shape from one reference frame to a block (square) of the current frame, thus obtaining a more general representation of camera and object motions, as described next.

There are several different motion models that can be used to warp blocks between frames, such as the affine, projective, and bilinear motion models [19]. In the past, these motion models have been used in the context of predictive video codecs to obtain more accurate predictions, especially when non-translational motion is present. For example, in [20], block matching with several first-order geometric transforms was proposed and significant RD performance improvements were obtained when compared to simple translational block matching. More recently, a parametric motion representation was proposed [21] where a Kanade-Lucas-Tomasi (KLT) tracking algorithm is adopted to match correspondence points and a motion segmentation algorithm is used to compute several parameter sets for a perspective transform; then, several warped reference pictures are generated and the best is selected for the coding process. In [22], a parametric Skip mode is proposed using a parametric motion estimation algorithm with cubic spline interpolation to obtain a set of motion model parameters for each frame; these parameters describe the global motion between the current and last decoded frames. Then, a new zero-residue Skip mode makes use of the estimated parameters to obtain a new prediction that can be selected for each block (although more typically used for background areas that may be well described with global motion). In the past, advanced motion models have brought significant advances in terms of coding performance for predictive video codecs at the cost of higher encoding and decoding complexity. However, these advanced motion models have not yet been exploited in the context of distributed video codecs, i.e., SI creation solutions employing advanced motion models based on geometric transforms to perform the SI frame interpolation are not available in the literature.

3. Bidirectional perspective side information creation: basics and architecture

The first task of a distributed video encoder based on the popular Stanford architecture is to classify the video frames into WZ frames and key frames; typically, the key frames are periodically inserted, determining the GOP size. While the key frames are Intra-encoded, this means without exploiting the temporal redundancy, the WZ frames are distributed encoded by using error correcting codes; for the WZ frames, the decoder generates the SI with the help of already decoded WZ and key frames, so-called *reference frames*. In this paper, a novel bidirectional perspective side information (BPSI) creation framework is proposed to model the motion between reference frames and obtain SI frames with improved quality. The proposed BPSI creation solution makes use of the perspective motion model by warping quadrilaterals from both (backward and forward) reference frames to estimate each SI block.

3.1 Perspective transform

Regarding the several motion models available in the literature, the eight-parameter perspective motion model was selected in this paper, due to its popularity [20-23] and also its ability to model complex motion, like zooms, rotations, and perspective deformations. The perspective transform provides a quadrilateral-to-quadrilateral mapping with the following representation:

$$[x, y, w] = [u, v, 1]\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad (1)$$

where $\{a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}\}$ are the perspective transform parameters, (u,v) an input quadrilateral vertex, and (x,y) the corresponding output warped quadrilateral vertex. To compute the eight perspective motion parameters, four pairs of correspondence points vertices, (u,v) and (x,y), are needed. Each pair of correspondence vertices is used to define a perspective transform vector, which is then used for the estimation of the best perspective transform model.

As usual, it is assumed that $a_{33} = 1$, while the eight a parameters are determined by solving the linear system in (2), using the four vertices of the input quadrilateral and the four corresponding vertices of the output quadrilateral:

$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \\ a_{13} \\ a_{23} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \qquad (2)$$

when square to quadrilateral mappings occur, as shown in Figure 1, the linear system in (2) can be simplified, leading to a lower complexity process.

After the perspective transform parameters, a, are found, any (u,v) point can be warped into a (x,y) point according to the perspective motion model using:

$$x = \frac{ua_{11} + va_{21} + a_{31}}{ua_{13} + va_{23} + 1} \qquad y = \frac{ua_{12} + va_{22} + a_{32}}{ua_{13} + va_{23} + 1} \qquad (3)$$

With (3), the warped coordinates (x,y) associated to the correspondence (u,v) coordinates can be calculated. Notice that motion models with fewer parameters are frequently employed in predictive video codecs since motion parameters have to be transmitted to the decoder, many cases with a significant impact on the final bitrate, thus significantly influencing the overall RD performance. However, in this paper, the motion model parameters are not transmitted since they are created and used at the decoder to create a SI frame, thus allowing more powerful motion representations without any bitrate penalty.

3.2 Walkthrough

The high-level architecture of the proposed BPSI creation solution is shown in Figure 2. To achieve the best SI quality and RD performance, the BPSI architecture includes two SI estimation branches: a conventional translational SI creation branch using the popular SI creation solution proposed in [9], and adopted in the DISCOVER DVC solution [8], and a novel advanced motion modeling branch exploiting the perspective transform characteristics.

Before describing in detail the BPSI architectural modules (see the next section), this sub-section presents a walkthrough of the full BPSI process to explain first in a concise way the overall high-level SI creation processing flow. In Figure 2, the shaded blocks correspond to the novel techniques proposed in this paper, notably with respect to the popular translational frame interpolation solution [9]. The BPSI creation framework still includes several translational techniques, notably backward motion estimation (ME), forward ME, bidirectional ME, and spatial motion filtering, which are implemented using conventional solutions [9]. The proposed BPSI architecture intends to exploit the best of the translational and perspective SI

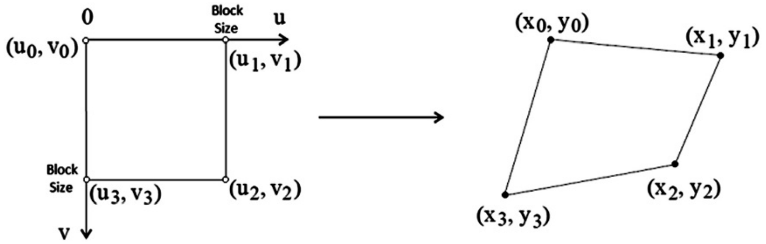


Figure 1 **Square to quadrilateral mapping example [20].**

creation approaches by selecting, at the block level, one of the approaches to generate the SI frame, thus adapting to the specific local content characteristics.

The conventional *BPSI translational motion branch* targets the creation of block-level SI candidates as follows:

1. *Backward ME* - Using both (previously decoded) reference frames, motion estimation with 16 × 16 block sizes and full pixel accuracy is performed for the forward direction, i.e., from the forward/future reference frame, X_f, to the backward/past reference frame, X_b. The motion estimation is performed using a weighted mean absolute difference (MAD) criterion [9] and provides a good starting point for the backward perspective transform search performed in step 7 and for the bidirectional ME technique described next. This process generates a *backward translational motion field*.
2. *Bidirectional ME (16 × 16 and 8 × 8)* - This step targets the refinement of the backward translational motion field already computed. With this purpose, the bidirectional translational ME is performed twice, first for 16 × 16 blocks and after for 8 × 8 blocks, always with the backward translational motion field to replicate the MCFI approach [9]. This technique incorporates several additional constraints in the translational motion field refinement, notably (1) all motion vectors must cross the center of each block in the SI frame and (2) the motion trajectories are restricted with an adaptive search range technique that defines the search windows by using information from neighboring blocks.
3. *Spatial motion filtering* - This technique spatially filters the noisy backward translational motion field obtained after the refinement in the previous step to reduce the number of 'incorrect' motion vectors when compared to the true motion field. The weighted vector median filter used [24] improves the motion field spatial coherence by identifying, for each SI block, the neighboring blocks motion vectors which can better represent the motion trajectory.

By performing the pure translational techniques in steps 1 to 3, and the motion compensation step to create the SI block at the end (step 12), the translational frame interpolation approach proposed in [9] is replicated. Naturally, for some of the SI frame blocks, the translational motion model 'fails' (in the sense that it provides poor SI quality), and a warped block estimated with a perspective motion model can provide higher SI quality.

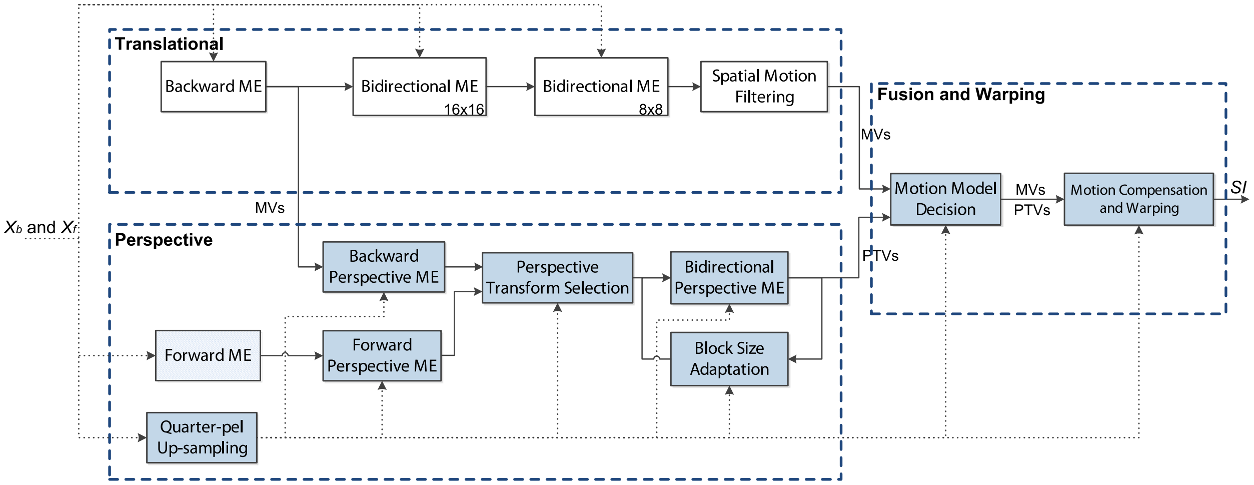


Figure 2 **Architecture of the proposed BPSI creation solution.**

The *BPSI perspective motion branch* targets the creation of SI candidate blocks as follows:

4. *Forward ME* - Here, step 1 is repeated for the forward direction, i.e., from X_b to X_f. Thus, a *forward translational motion field* is obtained to provide the starting point for the forward perspective transform search performed in step 6. Note that some of the motion vectors obtained in this step are not correlated in any way with the motion vectors from step 1, especially when occlusions or illumination changes occur, thus justifying the adoption of both the forward ME and backward ME steps.
5. *Quarter-pel up-sampling* - To provide a more accurate estimation for the possible perspective deformations, the backward and forward reference frames are first up-sampled with the H.264/AVC quarter-pel up-sampling filter [1]; this is performed so that the next steps can benefit from increased precision reference frames.
6. *Forward perspective ME* - This step receives as input the forward translational motion field estimated in step 4 to generate a *forward perspective motion field*. This motion field is a more complete representation of the motion between the reference frames as it includes, for each block, a set of four vectors, referred here as *perspective transform vectors* (PTVs), one for each vertex of the block; these vectors allow representing a whole range of deformations that cannot be obtained with a single motion vector as in the pure translational approach. Then, the up-sampled reference frames are used to estimate the best (in terms of distortion) perspective transform for each 16 × 16 block by searching for the deformation leading to the highest quality warped block; in this case, half-pel accuracy is used for the perspective transform vectors.
7. *Backward perspective ME* - Then, the process performed in the previous step is repeated for the opposite direction (from X_f to X_b), thus obtaining the perspective deformations (defined by the associated PTVs) from the future reference frame, X_f, to the backward reference frame, X_b. In this case, the backward translational motion field estimated in step 1 is used as input.
8. *Perspective transform selection* - This step aims at obtaining a reliable perspective transform for each SI frame block while avoiding holes and block overlappings that may occur in the SI frame. This module receives as input both the forward and backward perspective transforms (defined by their associated PTVs) and generates a unified perspective motion field for the SI frame, after eliminating the PTVs classified as unreliable (see Section 4 for more details).
9. *Bidirectional perspective ME* - Similarly to the translational SI creation solutions, bidirectional perspective motion estimation is performed with the perspective transforms selected in the previous step to refine the perspective deformations already obtained. This bidirectional perspective ME procedure is performed between the two reference frames while taking as reference the SI frame; it is performed twice, first with 16 × 16 blocks using as input the PTVs estimated in the previous step and after with 8 × 8 blocks after performing the adaptation described in the next step.
10. *Block size adaptation* - This step intends to estimate a perspective transform for 8 × 8 blocks using as input the perspective transforms obtained in the previous step for 16 × 16 blocks, i.e., the deformations of the 16 × 16 blocks are used to obtain the 8 × 8 block deformations. A perspective transform hierarchical approach is adopted because it was found that more accurate and coherent 8 × 8 block perspective transforms may be obtained from the corresponding 16 × 16 block perspective transforms than directly estimating the perspective transforms for 8 × 8 blocks.

After the two motion modeling branches have provided their best estimations, a motion model decision (step 11) that selects between the perspective and translational motion models is performed along with the final motion compensation and warping to fuse the available SI estimations (step 12). Thus, the final SI frame is created by the following steps:

11. *Motion model decision* - This step aims to choose, for each SI block, the best motion model between the translational model (characterized by *motion vectors*) and the novel perspective transform model (characterized by perspective transform vectors). This is a challenging task since the original frame is not available at the decoder to help assess the real quality (e.g., using a mean square error) of each of these SI estimations.
12. *SI creation* - Last, the final SI frame is created, following the decisions taken for each SI block in the previous step. Thus, for the blocks where the translational motion model was selected, the SI frame is created by motion compensation, while for the blocks where the perspective motion model was selected, warping of the quadrilaterals in both reference frames (followed by averaging) is made according to the perspective transforms obtained in step 9.

The perspective motion model used for SI creation (in a distributed video decoder) enables a more accurate characterization of complex motion that might occur in the video sequence, such as zooms, rotations, and other affine and perspective deformations, without transmitting any parameters or vectors from the encoder to the decoder as in predictive video coding. However, since the original frame is not available, this is a challenging task and several tools are necessary to compute and regularize the perspective transforms. In the next section, the details on the proposed techniques for the novel modules in the BPSI framework, notably exploiting the perspective motion model, are presented.

4. Bidirectional perspective side information creation: techniques

The novel algorithms proposed for the several modules in the BPSI architecture presented in Figure 2 are now described in detail in the following sub-sections. Naturally, more detail is provided for the algorithms related to the perspective motion modeling as they regard the major technical contributions of this paper.

4.1 Backward and forward motion estimation

This module receives as input the two (decoded) reference frames and estimates the translational motion field between those reference frames, without any information about the original frame. While backward ME is part of the translational MCFI technique [9], both the backward and forward ME motion vectors are used for the estimation of the best perspective motion model, thus the reason to explain them in detail here. However, since this technique is equivalent for the backward and forward directions, only the backward motion estimation process is described. The backward ME proceeds as follows:

1. *Identification of the reference frames* - Initially, the two relevant reference frames associated to the SI frame under estimation are identified. For a GOP size of 2, the reference frames are the two neighboring key frames of the interpolated SI frame, one in the past and another in the future. If a larger GOP size is used, previously decoded WZ frames are also used as reference frames while still using only the two neighboring reference frames; these neighboring frames are defined as proposed in [9].
2. *Motion estimation* - After, the two reference frames are low-pass filtered to obtain a more spatially coherent motion vector field. Motion estimation is then performed from X_f to X_b, i.e., in the backward direction. For this, a block matching algorithm employing a modified matching criterion is used [9]; the modified criterion adds weights to the MAD criterion to favor translational motion vectors closer to the block center to regularize the translational motion field. The block size is 16 × 16 and full pixel accuracy is used.

This backward motion vector field is later refined with bidirectional motion estimation and spatial motion smoothing techniques; please refer to [9] for more details.

4.2 Backward and forward perspective motion estimation

The main objective of this technique is to estimate the perspective motion, locally, notably for each block of the backward and forward reference frames, using as starting point the translational motion vectors obtained from the backward and forward (translational) motion estimation processes previously described. Only the backward perspective motion estimation performed between the X_b to X_f decoded reference frames is described here as the forward estimation is equivalent. The architecture of the proposed perspective motion estimation technique is shown in Figure 3; several key tasks are performed here, namely the estimation of the perspective motion model parameters and the selection of the best perspective transform, i.e., the perspective transform that creates (by warping) the highest quality SI block.

The following steps are performed to obtain the best perspective backward transform for each 16 × 16 X_f block to be characterized by the selected PTVs:

1. *Perspective transform initialization* - In this step, the initialization of the perspective transform estimation is performed by simply assigning to the four corners of each block in the forward reference frame the (same) motion vector calculated with the translational backward motion estimation algorithm, thus obtaining the initial perspective transform vectors. In this deformation, the warped quadrilateral corresponds to a square block (see Figure 4a) in the same position and size as the displaced block calculated by (translational) motion estimation.
2. *Vertex selection* - A block vertex is selected for processing, starting with the upper left vertex and following a clockwise direction. To estimate the best perspective transform, a possible solution would be to evaluate every possible combination of PTVs; for example, with a 32 × 32 pixel search window located at each vertex, it would be necessary to estimate a prohibitive number of transforms for each block. Since the full-search complexity is rather high, a novel search algorithm is proposed to find the best PTVs for each block, providing a good trade-off between SI quality and perspective modeling complexity. Thus, the 'best' perspective transform is estimated by evaluating the PTV associated to each block vertex individually while keeping the remaining three PTVs (associated to the remaining

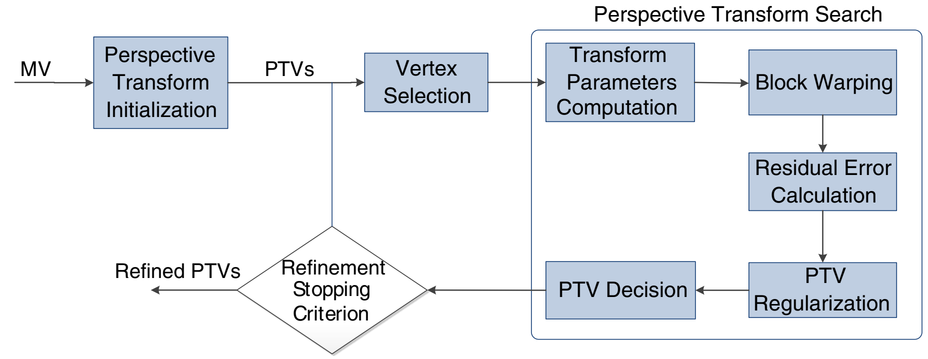

Perspective Transform Search
MV
Perspective Transform Initialization
PTVs
Vertex Selection
Transform Parameters Computation
Block Warping
Residual Error Calculation
Refined PTVs
Refinement Stopping Criterion
PTV Decision
PTV Regularization

Figure 3 **Architecture of the proposed forward/backward perspective motion estimation modules.**

three vertices) in a fixed position, as illustrated in Figure 4b.

3. *Perspective transform search* - In this step, the perspective transform of each X_f block is found by evaluating several possible deformations, i.e., the quality of several warped blocks is evaluated. For each X_f block, the search for the best perspective transform in the past X_b frame is made as follows:
   a. Transform parameter computation: After selecting a vertex and setting its initial search point within the search window, the eight perspective parameters for the quadrilateral deformation in the backward reference frame are obtained considering each (square) block in the forward reference frame. In such case, the parameters are obtained by simply solving the linear system in (2) using the four vertices of the quadrilateral and the four vertices of the corresponding square block.
   b. Block warping: In this step, a warped square block in frame X_f is generated (using the corresponding quadrilateral in frame X_b) with the perspective parameters calculated in the previous step. In such case, (3) can be used to obtain the warped coordinates (x,y) associated to the corresponding (u,v) points inside each square block. This procedure is shown in Figure 5, where the quadrilateral in frame X_b is warped to a square block in the current frame X_f (left square block), where a reference square block (right block) is already available. The mapping of a regular square grid of pixels (X_f block) into a quadrilateral (in X_b) usually leads to positions with non-integer coordinates. Thus, it is necessary to have a reliable method to estimate the pixel values for these warped positions. To achieve this target, the first step is to up-sample the reference frames with the quarter-pel H.264/AVC motion compensation interpolation filter to provide more precise interpolated values for the warped positions. However, since a position with any arbitrary (real value) precision may be obtained, the obtained quarter-pel samples are not enough. Therefore, a bilinear interpolation

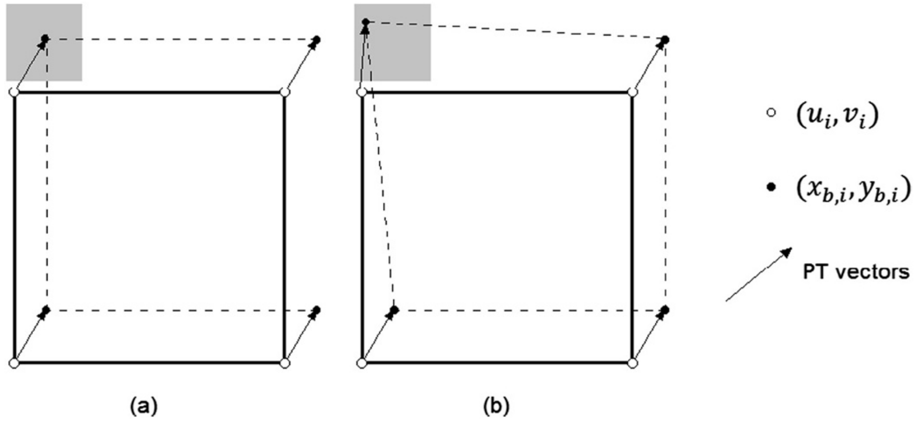


Figure 4 **PTV search: (a) initial positions and (b) possible PTV positions after performing the search for the first vertex.**

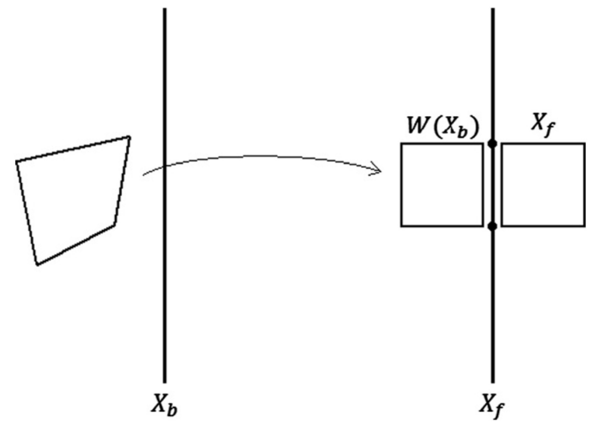


Figure 5 **Blocks compared during the backward perspective warping.**

method is used to estimate a pixel value at any warped position based on the quarter-pel samples; this interpolation filter is formalized in (4) and illustrated in Figure 6:

$$P(x,y) = (1{-}L_x)\left(1{-}L_y\right)P_a + L_x\left(1{-}L_y\right)P_b + (1{-}L_x)L_yP_d + L_xL_yP_c \tag{4}$$

As shown in Figure 6, the estimation for a pixel value P in any arbitrary position (x,y) is based on L_x and L_y, which represent the distances to the quarter-pel positions in the square grid, both horizontally and vertically. In (4), $P(x,y)$ is obtained by averaging the four neighboring quarter-pel pixel values P_a, P_b, P_c, and P_d weighted by their distance to $P(x,y)$. After some manipulation, the following expression is obtained:

$$P(x,y) = (P_b{-}P_a)L_x + (P_d{-}P_a)L_y + (P_c{-}P_d{-}P_b + P_a)L_xL_y + P_a \tag{5}$$

For a perspective transform under evaluation, all the X_f block pixel positions can be projected into frame X_b by computing (5), and their interpolated values can be obtained, i.e., the warped candidate block (block W (X_b) in Figure 5) can be created.

c. Residual error calculation: To evaluate the quality of the warped block calculated in the previous step, a MAD metric is adopted. In this case, the residual error calculation is performed between the warped block W_b (from reference frame X_b) and the corresponding reference block in frame X_f (represented by two square blocks in Figure 5):

$$\mathrm{MAD}(j) = \frac{1}{N \times N}\sum_{(x,y)\in B_j}\left|X_f(x,y){-}W_b(x,y)\right| \tag{6}$$

where N is the block size and B_j is the jth block in the reference frame X_f.

d. PTV regularization: Since the original frame is not available, to obtain a perspective transform that is closer to the true motion of the objects/blocks in the video sequence [9], it is necessary to apply a regularization technique. Thus, it is not enough to minimize the MAD residual error as in (6), as many motion estimation solutions do, but it is also necessary to avoid large deviations between neighboring motion models. In this case, a simple regularization criterion, favoring the PTVs closer to the origin and avoiding PTVs with large magnitudes (which most likely do not represent true motion) is used. This PTV regularization consists in applying a penalty to the MAD obtained in the previous step, directly proportional to the distance between the current and initial PTVs, i.e., the PTV available as input to this module. The proposed process includes two steps: first, a *local/vertex weighting regularization* and after a *global/block weighting regularization.* The proposed *local weighting regularization* technique computes the distortion D_l associated to each warped block as:

$$\delta_l(x,y) = \sqrt{(x{-}x_c)^2 + \left(y{-}y_c\right)^2} \tag{7}$$

$$D_l = \mathrm{MAD}(1 + k\delta_l) \tag{8}$$

where (x_c,y_c) represents the initial PTV position for a given vertex, δ_l represents the distance between the current PTV position (x,y) and (x_c,y_c) for the local weighting approach, and k is a scaling factor. In this case, the distortion D_l is regularized with a cost that is directly proportional to the distance between the current PTV and the initial PTV of that vertex.

After an estimate of the four PTVs (one for each vertex) for the block is available, a *global weighting regularization* is applied. In this approach, the PTVs obtained for all vertices are globally refined, i.e., the penalty applied to the MAD is directly proportional to the sum of the distances between the current PTV positions and the corresponding initial PTV positions for the

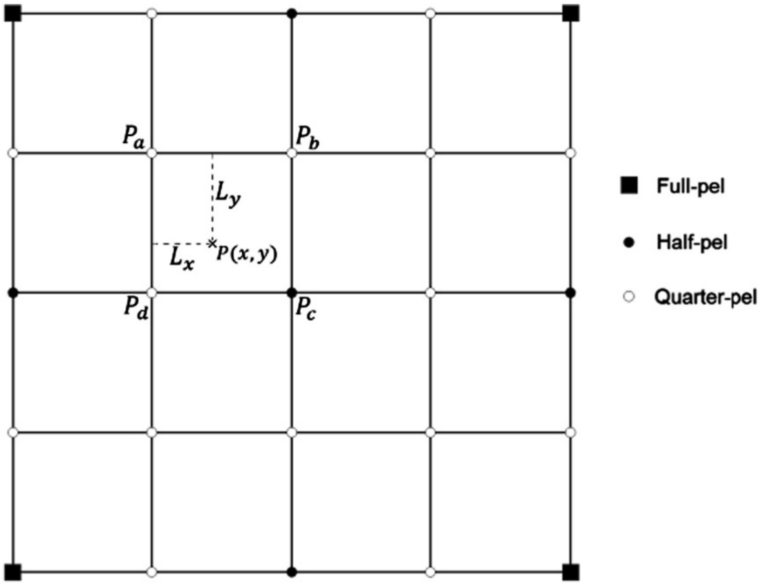


Figure 6 **Bilinear pixel interpolation for a regular square grid.**

four vertices. For the global weighting approach, the distortion D_g of each block is computed as:

$$\delta_g = \sum_{i=1}^{4} \delta_l(x_i, y_i) \tag{9}$$

$$D_g = \mathrm{MAD}\left(1 + k\frac{\delta_g}{4}\right) \tag{10}$$

where δ_g represents the sum of the distances of the PTVs obtained after local weighting regularization to (x_c,y_c), calculated independently for each vertex. It was found experimentally that k could be the same for both the global and local weighting regularization approaches as no benefits were obtained with different k values. Finally, steps a to d are repeated until all positions inside the PTV search range are tested.

e. PTV decision: From all the PTVs evaluated inside the search window, the PTV leading to the minimum D_l (only when at least one vertex has not yet been processed for regularization) or D_g is selected. Then, steps 2 and 3 are repeated for each of the remaining three block vertices, using a clockwise rotation. When the four vertices have been processed and the corresponding PTVs are obtained, a full refinement iteration is completed, and the algorithm proceeds to step 4.

4. *Refinement stopping criteria* - If the PTVs do not change during a complete refinement iteration, the algorithm stops the PTV search process, implying that the search algorithm has successfully converged to a solution, i.e., the best PTVs have been found for the block under processing. Otherwise, the number of iterations is incremented and the algorithm goes back to step 2. For most cases, the computational complexity can be reduced when compared to an approach where a fixed number of iterations are executed. Anyway, to stop the search algorithm for the cases where convergence is difficult to obtain, the adopted maximum number of iterations is five.

4.3 Perspective transform selection

The perspective transforms obtained with the technique proposed in the previous section should represent well the motion of each X_b or X_f block; however, the frame to estimate is Y_i for which there is still no perspective transform available. Thus, the perspective transform selection technique aims to obtain reliable perspective transforms for the SI frame using the perspective transforms previously estimated, this means for both the forward and backward directions, and only involving the reference frames X_b and X_f. In the past, the exploitation of two SI estimations according to the direction (backward vs. forward) was also adopted to handle sequences where occlusions and complex motion occur [18]. This process includes three main steps: first, the backward and forward perspective transforms considered unreliable are eliminated; second, for each SI block, two perspective transforms (one from each direction) are selected; and, third, only one transform is chosen as the final perspective transform for the SI block after evaluating both the available transforms.

Considering T as a group of four PTVs, one for each vertex, representing the perspective deformation of a given X_b or X_f block, the process to obtain a perspective transform for each SI block proceeds as follows:

1. *Perspective transform filtering* - This step compares the two residual (MAD) errors obtained with the forward perspective transform, T_f, and the backward perspective transform, T_b, obtained for the same block position, but in their respective reference frames. This comparison is performed to decide if both perspective transforms are kept or one of them is excluded, trying to filter out perspective transforms that do not potentially lead to good SI quality. The filtering decision performed for each block is done according to the following rules:

$$\begin{array}{ll} \text{No filtering} & \text{if } \left|\mathrm{MAD}_b - \mathrm{MAD}_f\right| < \tau \\ T_f \text{ removed} & \text{if } \left|\mathrm{MAD}_b - \mathrm{MAD}_f\right| \geq \tau \wedge \mathrm{MAD}_b < \mathrm{MAD}_f \\ T_b \text{ removed} & \text{if } \left|\mathrm{MAD}_b - \mathrm{MAD}_f\right| \geq \tau \wedge \mathrm{MAD}_f < \mathrm{MAD}_b \end{array} \tag{11}$$

   where MAD_b is the residual error calculated between a given block in X_f and the respective block obtained with the backward transform T_b (similarly for MAD_f), and τ is a threshold. If $|\mathrm{MAD}_b - \mathrm{MAD}_f| < \tau$, both the backward transform, T_b, and the forward transform, T_f, are considered reliable; thus, no transform is eliminated. When $|\mathrm{MAD}_b - \mathrm{MAD}_f|$ is larger or equal than τ, one of the transforms is considered more reliable than the other and two cases are considered: (1) if $|\mathrm{MAD}_b < \mathrm{MAD}_f|$, T_b is kept (MAD_b has the lowest value) and T_f is dropped; and (2) if $\mathrm{MAD}_f < \mathrm{MAD}_b$, T_f is kept (MAD_f has the lowest value) and T_b is dropped.
2. *Perspective transform selection* - Now, the best transform for each SI block has to be selected from all the transforms T_b and T_f considered reliable in the previous step, i.e., the T_b and T_f perspective transforms that were not eliminated. Note that these transforms do not characterize the perspective deformation associated to a SI block but the deformations of blocks in the backward and forward reference frames. For this selection, a simple criterion based on the distance between the position where each PTV intersects the SI frame and the corresponding SI block vertex is used. First, two perspective transforms, $\dot{T}_b$ and $\dot{T}_f$ for each SI block are selected, one for each backward/forward direction; then, both are evaluated in terms of residual error and the best transform is selected. From all the backward perspective transforms, T_b, only one is selected for each SI block (denoted as $\dot{T}_b$); the following selection procedure is made for each SI block:
   a. For each PTV associated to a given transform, T_b, the distance d_i between the corresponding SI block vertex and the point where the PTV intersects the SI frame (see Figure 7) can be calculated as:

$$d_i = \sqrt{\left(x_{b,i} + \left(x_{f,i} - x_{b,i}\right)\bar{d}_b - u_i\right)^2 + \left(y_{b,i} + \left(y_{f,i} - y_{b,i}\right)\bar{d}_b - v_i\right)^2} \tag{12}$$

$$\bar{d}_b = \frac{d_b}{d_b + d_f} \tag{13}$$

   In Figure 7, $(x_{b,i}, y_{b,i})$ and $(x_{f,i}, y_{f,i})$ are the PTV positions of vertex i in X_b and X_f, respectively, $\bar{d}_b$ is the normalized distance of the SI frame to the backward frame defined in (13), and (u_i, v_i) are the vertices of the SI block under consideration.
   b. Then, the overall distance, d_T^b, considering the four vertices of each SI block, is computed for T_b using (14) and defines the overall distance between the positions where the transform T_b intersects the SI frame and the corresponding SI block for which no perspective transform is yet available.

$$d_T^b = \sum_{i=1}^{4} d_i \tag{14}$$

   c. After obtaining the overall distance d_T^b for each T_b transform, the T_b leading to the minimum overall d_T^b distance is selected, obtaining the perspective transform $\dot{T}_b$ for the SI block under consideration. Then, the previous steps a to c are repeated for the forward direction to find the best forward transform, $\dot{T}_f$, for the same SI block.
3. *SI perspective transform creation* - In this step, the deformation for each SI block (and not anymore of

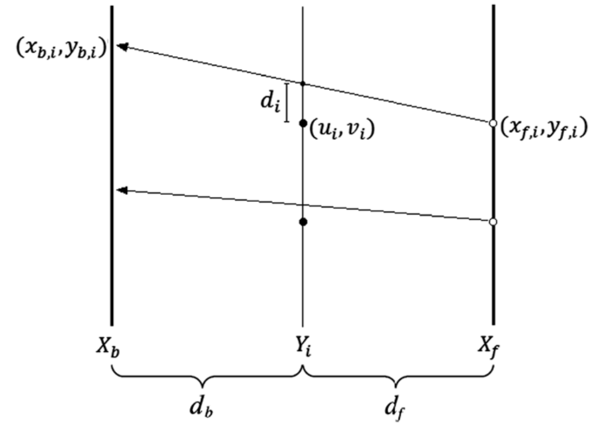


Figure 7 Graphical representation of the distance d_i.

the blocks in the reference frames) is found by selecting just one perspective transform. Thus, the PTVs of the two transforms, $\dot{T}_b$ and $\dot{T}_f$, obtained in the previous step for each SI block, are displaced, so that the PTVs cross the respective vertex on the SI block. With the selected perspective transforms, $\dot{T}_b$ and $\dot{T}_f$, the procedure illustrated in Figure 8 is applied for each SI block:

a. First, a block warping procedure is applied where two warped blocks are generated using two sets of perspective transform parameters, derived (by solving (2)) from each of the perspective transforms $\dot{T}_b$ and $\dot{T}_f$, one for each up-sampled reference frame (backward and forward). The two warpings, W, obtained for each transform, correspond to two sets of perspective transform vectors, one set of vectors pointing from the WZ frame to X_b and another pointing from the WZ frame to X_f. Then, the MAD metric in (6) is applied to calculate the residual error, as for the backward and forward perspective ME; the only difference is that the MAD (represented as the minus sign in Figure 8) is calculated between the two warped blocks, W_b^1 and W_b^0 for transform $\dot{T}_b$ and W_f^1 and W_f^0 for transform $\dot{T}_f$.
b. Then, the perspective transform T_s leading to the minimum MAD is selected, i.e., T_s is made equal to $\dot{T}_b$ or $\dot{T}_f$ depending on which of these transforms leads to the minimum MAD. The new perspective transform, T_s, represents the motion between the SI block and the corresponding reference frames, X_b and X_f.

For the blocks at the SI frame border, a unidirectional motion compensation mode is adopted if the corresponding quadrilateral has more than 25% of the block area outside the reference frame border. In such case, the estimation obtained from the reference is considered unreliable, and block warping is only performed with the remaining reference frame. In the rare case where both quadrilaterals have more than 25% of the corresponding area outside the reference frame, the bidirectional mode, as for the other non-border blocks, is still used.

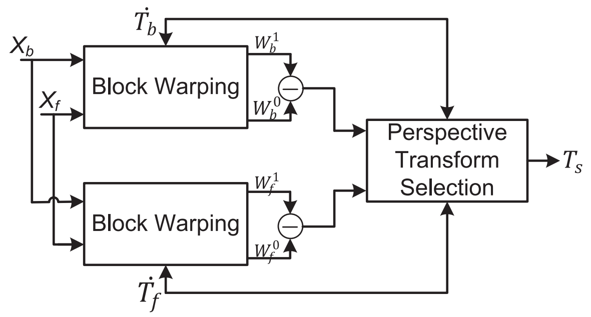


Figure 8 SI perspective transform creation architecture.

4.4 Bidirectional perspective transform estimation

This module is applied twice, first with a 16 × 16 block size and after with a 8 × 8 block size, and aims to refine the perspective transform (only one) obtained for each SI block in the previous step. A hierarchical approach was adopted so that the 16 × 16 perspective transforms can provide a good starting point for the final 8 × 8 perspective transforms. The architecture of this module is similar to the backward and forward perspective ME technique presented in Section 4.2. The major difference is that this module receives as input a perspective transform for each SI block and no longer translational motion vectors (as in the forward and backward perspective transform estimation modules) and refines the initial transforms to obtain better SI quality. This module is also able to correct some of the errors and inaccuracies made in the algorithm presented in the previous section which selects and creates perspective transforms for the SI blocks based on the perspective transforms obtained between the reference frames. To obtain a refined set of four PTVs for each SI block, the algorithm presented in Section 4.2 is applied. The major difference is that the linear trajectory of the PTVs for each block vertex must be preserved, i.e., the forward PTV position needs to be symmetric to the backward PTV position considering the (u_i,v_i) vertex in the SI frame, and the PTV has always to cross the vertex (u_i,v_i) in the interpolated SI frame. This constraint is similar to the constraint considered in the translational bidirectional motion estimation algorithm already proposed in [9]. In addition, the constrained block warping procedure described in the previous section is also applied: from a candidate SI perspective transform, two sets of transform parameters are calculated to describe the motion between the SI frame and both the backward and forward reference frames. Then, two warped blocks are obtained for each SI block and the MAD residual error is computed to evaluate the quality of each candidate perspective transform. By displacing the PTVs, other candidate perspective transforms can be tested with an iterative search procedure to find the best perspective transform (as in Section 4.2).

4.5 Block size adaptation

The main objective of this module is to obtain more precise PTVs (finer scale) based on the PTVs already obtained for a coarser scale. Thus, the PTVs for the four 8 × 8 blocks in each 16 × 16 block are computed by applying the perspective transform of the corresponding 16 × 16 block. This procedure is performed for each SI block and for both the backward and forward PTVs. The final result for one of the directions is shown in Figure 9.

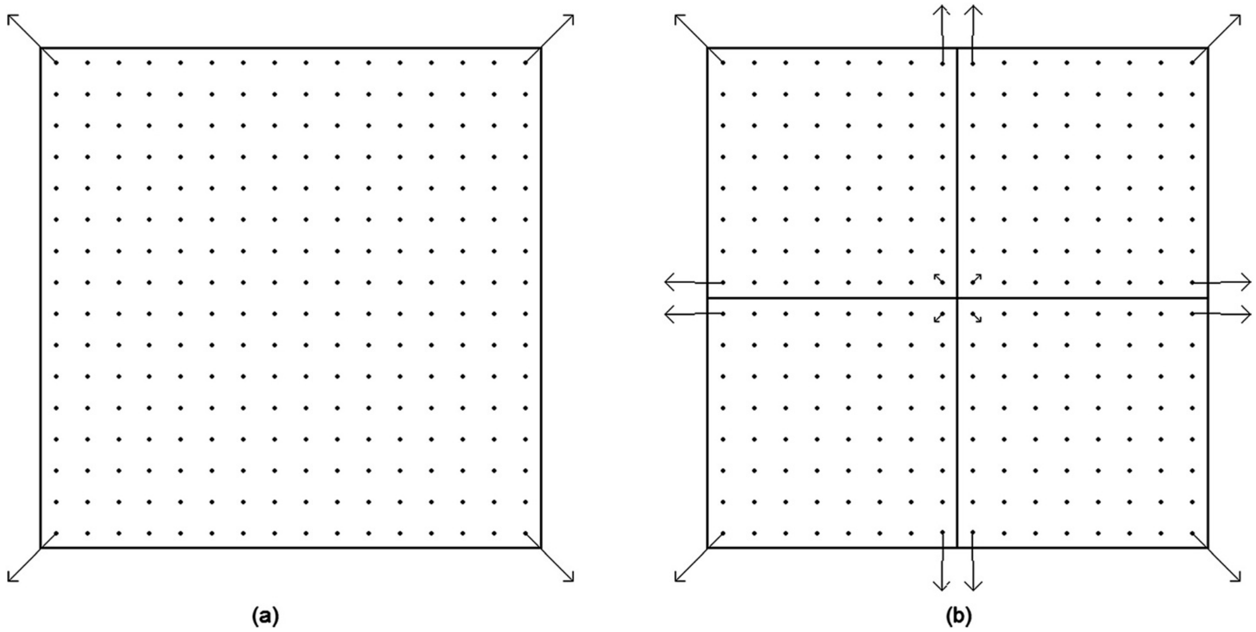


Figure 9 Block size adaptation example for one direction. (a) PTV positions for a 16 × 16 block and **(b)** obtained PTV positions for the four smaller 8 × 8 blocks.

Thus, the perspective model parameters, a, already computed for the 16 × 16 blocks by solving the linear system in (2), are used to obtain the PTVs for a finer 8 × 8 scale. More precisely, for every new vertex (in the 8 × 8 block), the corresponding projection in the reference frame is found by using (3), thus obtaining four sets of PTVs corresponding to the four 8 × 8 blocks in each 16 × 16 block.

4.6 Motion model decision

The motion model decision algorithm selects, for each SI block, the best motion model (translational or perspective) using a MAD criterion, i.e., the motion model with the minimum MAD residual error for a given SI block is chosen. The MAD_t residual error for the translational mode has already been obtained when performing the bidirectional translational ME for the 8 × 8 blocks [9], while the MAD_p residual error for the perspective mode was obtained when performing the bidirectional perspective ME algorithm for the 8 × 8 blocks. Regarding this decision, it was possible to observe that, for a significant number of SI blocks, the perspective mode could have a slightly better MAD value than the translational mode without leading to a final better SI estimation. Based on these observations, it is proposed to apply a penalty offset to the perspective mode MAD to ensure that this mode is only selected when it is reasonably better than the translational mode, thus increasing the probability of obtaining better SI quality. For each SI block, the motion modeling mode, ϕ_{si}, is calculated according to:

$$\phi_{\mathrm{si}} = \begin{cases} \mathcal{P}, \text{if } \ \mathrm{MAD}_p < \mathrm{MAD}_t - \alpha \\ \mathcal{T}, \text{if } \ \mathrm{MAD}_p \geq \mathrm{MAD}_t - \alpha \end{cases} \qquad (15)$$

where α is the penalty offset, ϕ_{si} represents the selected motion modeling mode for each SI block, and $\mathcal{T}$, $\mathcal{P}$ represent the translational and perspective motion modeling modes, respectively.

4.7 Motion compensation and warping

Finally, the SI frame is created according to the motion model previously selected for each SI block. The calculated perspective motion model parameters, the bilinear interpolation method, and the up-sampled reference frames are used to obtain the warped pixel values for the relevant SI block when the perspective mode is selected. When the translational mode is used, only the motion vector previously calculated (i.e., after spatial motion filtering) is used to obtain the SI block. In both cases, two SI estimations are available, one using each of the backward and forward reference frames, respectively, and the final SI block is obtained by averaging, followed by rounding, the warped or motion compensation SI estimations according to:

$$\mathrm{SI} = \begin{cases} \lfloor \bar{d}_b W_f + W_b (1 - \bar{d}_b) + 0.5 \rfloor \text{ if } \phi_{\mathrm{si}} = \mathcal{P} \\ \lfloor \bar{d}_b P_f + P_b (1 - \bar{d}_b) + 0.5 \rfloor \text{ if } \phi_{\mathrm{si}} = \mathcal{T} \end{cases} \qquad (16)$$

where W_b and W_f are the warped blocks obtained from the backward and forward reference frames, P_b and P_f

the motion compensated blocks obtained from the backward and forward references frames, $\bar{d}_b$ the normalized temporal distance already defined in (13), and $\lfloor\,\rfloor$ the floor operator.

5. Bidirectional perspective side information creation: performance evaluation

After presenting the proposed perspective transform motion modeling-based SI creation solution, it is time to assess the performance of the BPSI framework in terms of both SI quality and RD performance in the context of a state-of-the-art DVC codec. In this context, the next sub-section provides first a brief description of the DVC codec employed for RD performance evaluation and its encoding and decoding tools.

5.1 DVC-BPSI video codec

The proposed BPSI creation solution is used in the context of the DVC-BPSI codec which is a state-of-the-art DVC solution following the Stanford DVC architecture originally proposed in [18]. A simplified version of the DVC-BPSI codec architecture is shown in Figure 10.

The proposed DVC-BPSI codec corresponds to the DVC codec proposed in [25] taking the proposed BPSI creation framework as the SI creation solution. To obtain a powerful DVC solution, the DVC-BPSI codec includes state-of-the-art DVC coding techniques available in the literature. In summary, the DVC-BPSI encoder includes the H.264/AVC 4 × 4 DCT transform, a uniform scalar quantizer and a LDPC syndrome code as the Slepian-Wolf codec. The DVC-BPSI decoder uses a CRC code for error detection, a minimum mean square error reconstruction method, and an offline Laplacian correlation noise model at band level [26]. In the following, the DVC-BPSI codec is compared to a DVC-MCFI codec corresponding to the same DVC solution but using the popular MCFI SI creation solution [9] instead of BPSI, i.e., using a pure translational motion model for SI creation.

5.2 Test conditions

To evaluate the SI quality and the overall RD performance, meaningful and precise test conditions must be first defined. The DISCOVER project [8] has provided a detailed, clear, and complete set of test conditions that are currently widely used in the DVC literature; thus, the test conditions used for the evaluation of the proposed SI creation solution are similar to the DISCOVER test conditions.

5.2.1 Video sequences

To evaluate the proposed DVC-BPSI solution, four video sequences were selected, with different characteristics, notably in terms of motion and texture, thus leading to a rather complete and meaningful set of sequences and results. The selected video sequences are *Hall Monitor*, *Mobile and Calendar*, *Bus*, and *Stefan.* In Figure 11, a representative frame of each sequence is presented.

In Table 1, the characteristics of each video sequence are presented, notably the spatial and temporal resolutions as well the total number of frames for each sequence.

5.2.2 Coding parameters

The coding parameters and configurations used to evaluate the DVC-BPSI performance are as follows:

- *GOP size* - As common in the DVC literature, a fixed GOP size of 2 was adopted.
- *Key frames coding* - The key frames are coded with H.264/AVC Intra in the Main profile since this is one of the best Intra coding schemes in terms of RD performance.
- *Quantization parameters* - To perform the experimental evaluation, eight RD points (Q_i) were defined in terms of the H.264/AVC Intra key frame quantization parameter (QP_I) and the quantization level matrix for the WZ frames. The WZ frame quantization level matrices define for which DCT bands parity bits are transmitted with each matrix entry indicating the number of quantization levels for the respective DCT band (0 means that no parity bits are transmitted for that DCT band). The eight quantization level matrices considered here for RD performance evaluation are presented in Figure 12. The decoded quality (after reconstruction) depends on the quantization level matrix chosen and the corresponding key frame quantization step.

The key frame QP_I values were defined using an iterative process, which stopped when the average WZ frame PSNR was similar to the average key frame PSNR to avoid significant temporal quality variations which may have a negative user impact. The QP_I value selection procedure for the key frames assures an almost constant decoded video quality for the full set of frames (key frames and WZ frames) which is essential from the subjective quality point of view. Notice that distributing the same total bitrate in a different way between WZ and key frames may even lead to better RD performance, for example, by investing more bits in the key frames at the cost of a less stable video quality, but the resulting strong quality variations along time are highly undesirable. In Table 2, the QP_I values used for each RD point and each video sequence are presented.

- *MCFI parameters* - In the backward ME and bidirectional ME modules, the parameters are chosen according to [9], namely using a scaling factor $k = 0.05$ and a 32 × 32 search window.

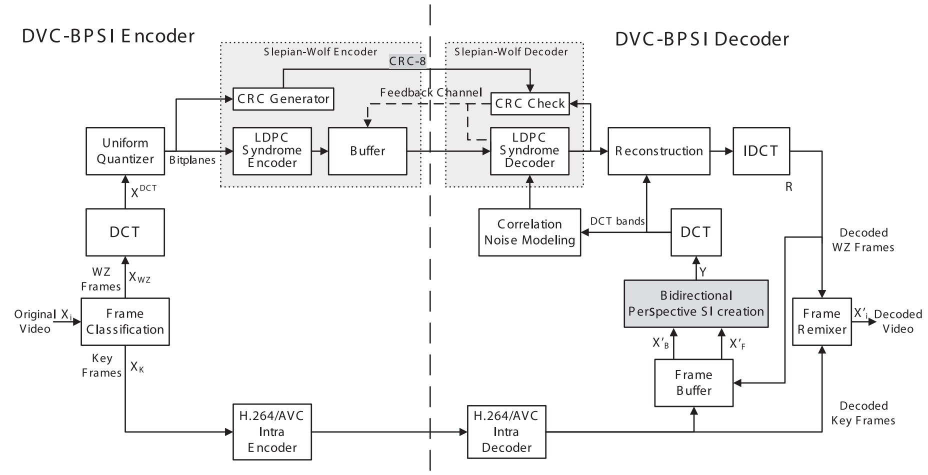


Figure 10 Simplified DVC-BPSI codec architecture.

- *BPSI parameters* - The forward and backward perspective ME is performed with a 9 × 9 search window for each vertex. The bidirectional perspective ME for the 16 × 16 blocks is performed with half pixel accuracy PTVs and a 7 × 7 search window for each vertex, while the 8 × 8 blocks use quarter pixel accuracy PTVs and a 5 × 5 search window for each vertex. For the bidirectional perspective ME, the scaling factor for the 16 × 16 blocks, k_{16}, is 0.05, while for the 8 × 8 blocks, k_8 is 0.21. To perform the motion model decision, a penalty offset, α, equal to 1 was applied to the MAD obtained with the perspective motion model. The BPSI parameters were defined following extensive experiments using as criterion the maximization of the SI quality gains. The video sequences used to compute the suggested parameter values to assess the RD performance are Mobile and Calendar, Hall Monitor, Stefan, Bus, Soccer, Container, Table Tennis, Coastguard, and Foreman at Quarter Common Intermediate Format (QCIF) spatial resolution.

5.2.3 Coding benchmarks

To evaluate the performance of the proposed DVC-BPSI video codec, the RD performance is compared to some relevant benchmarking solutions, notably the H.264/AVC Intra, H.264/AVC Zero Motion, and DVC-MCFI codecs. These video coding solutions share an important characteristic: all the encoders under evaluation have rather low encoding complexity (although not necessarily precisely the same) as they do not use motion estimation at the encoder. More precisely, the video coding solutions used as benchmarks are as follows:

- *H.264/AVC Intra* - H.264/AVC Main profile video codec using only the Intra mode as this is one of the most powerful and efficient Intra video codecs.
- *H.264/AVC Zero Motion* - H.264/AVC video codec exploiting only some temporal redundancy as no motion estimation is performed at the encoder. The same prediction structure adopted for the DVC-BPSI codec is used, i.e., a GOP size of 2, with an IBI GOP

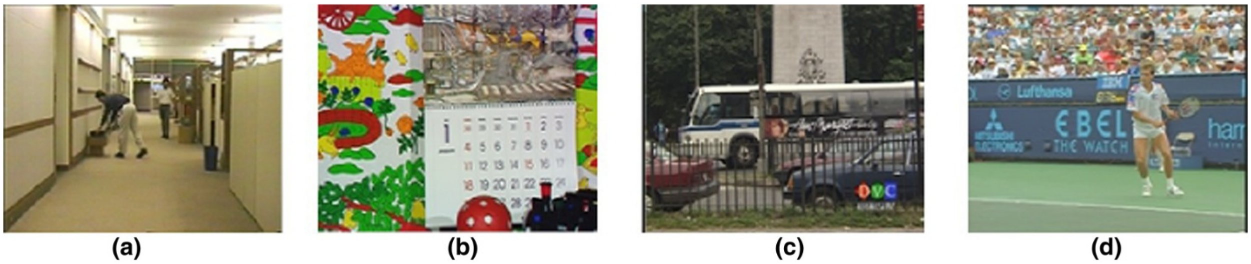


Figure 11 Test video sequences. (a) Frame 63 of Hall Monitor, **(b)** first frame of Mobile and Calendar, **(c)** frame 29 of Bus, and **(d)** first frame of Stefan.

Table 1 Test video sequences characteristics

Video sequence	Luminance spatial resolution	Temporal resolution [Hz]	Total number of frames
Hall Monitor	176 × 144	15	165
	352 × 288	30	330
Mobile and Calendar	176 × 144	15	149
	352 × 288	30	300
Bus	176 × 144	15	75
	352 × 288	30	150
Stefan	176 × 144	15	149
	352 × 288	30	300

structure and two reference frames, one in the past and another in the future.

- *DVC-MCFI* - To also assess the DVC-BPSI performance regarding an alternative state-of-the-art DVC codec, the DVC-MCFI codec performance is also used as benchmark. The only difference between the DVC-MCFI video codec and the proposed DVC-BPSI codec is the technique used in the SI creation module. The DVC-MCFI codec is rather similar to the DISCOVER DVC codec: the only differences regard the LDPC syndrome codec [27] adopted in the DVC-MCFI codec and the sub-pel motion vector accuracy adopted in the DISCOVER DVC codec.

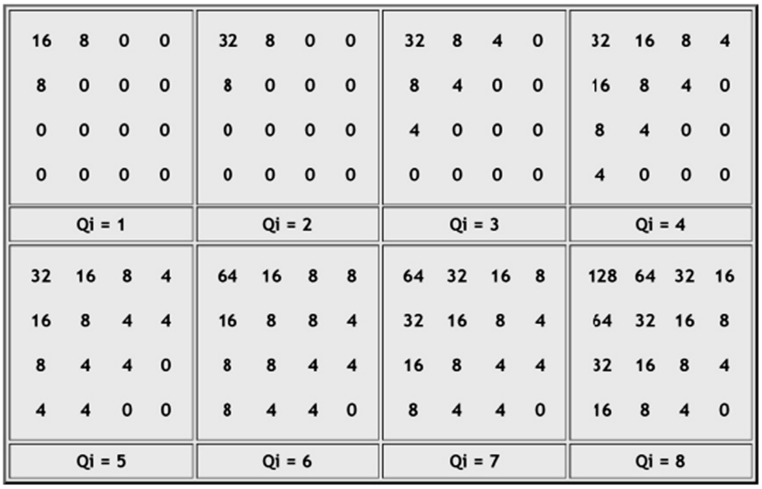
For all codecs under evaluation, only the luminance component was coded, meaning that the SI quality and the RD performance consider only the luminance rate and quality (naturally, for both the key frames and WZ frames). For the DVC-MCFI and DVC-BPSI key frame coding and the H.264/AVC Intra and H.264/AVC Zero motion codecs, the Main profile was selected (as typically in the DVC literature) since it allows high RD performance, even if with some encoding complexity cost associated to the CABAC entropy encoder and Intra coding modes.

5.3 Side information performance evaluation

The main target of this section is to evaluate the SI quality obtained with the proposed BPSI solution standalone, i.e., before integration in any DVC codec. Thus, the BPSI SI quality is only compared to the state-of-the-art MCFI SI quality for all the test sequences. The average BPSI and MCFI SI qualities for the whole sequences are shown in Tables 3 and 4 for two RD points, where Δ_{BWSI} expresses the average BPSI SI PSNR gain regarding the MCFI solution (value in italic corresponds to the highest PSNR gain). This evaluation also allows comparing the SI PSNR gains with the complete RD performance gains, which are presented in the next section for the proposed DVC-BPSI codec.

From the results in Tables 3 and 4, the following conclusions may be derived:

- The most significant SI quality gains are obtained when the key frame quality is better, i.e., when a lower QP is used (corresponding to the RD point Q_7). In fact, the lower gains obtained for the RD point Q_2 can be easily explained by the poorer quality of the key frames, which limits the SI gains, as it is difficult to obtain accurate perspective transforms when the reference frame quantization error is too high.
- The proposed BPSI solution shows more significant gains for the Mobile and Calendar video sequence. This can be explained by the camera motion present



16	8	0	0	32	8	0	0	32	8	4	0	32	16	8	4
8	0	0	0	8	0	0	0	8	4	0	0	16	8	4	0
0	0	0	0	0	0	0	0	4	0	0	0	8	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
Qi = 1				Qi = 2				Qi = 3				Qi = 4			
32	16	8	4	64	16	8	8	64	32	16	8	128	64	32	16
16	8	4	4	16	8	8	4	32	16	8	4	64	32	16	8
8	4	4	0	8	8	4	4	16	8	4	4	32	16	8	4
4	4	0	0	8	4	4	0	8	4	4	0	16	8	4	0
Qi = 5				Qi = 6				Qi = 7				Qi = 8			

Figure 12 Number of quantization levels associated to each DCT coefficient band for the eight RD points. Zero means that the DCT band is left uncoded, i.e., no parity data is transmitted [8].

Table 2 Key frame quantization parameters for each RD point, Q_i

Video sequences	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
Hall Monitor	37	36	36	33	33	31	29	24
Mobile and Calendar	38	37	36	35	35	34	32	28
Bus	45	44	43	39	39	37	34	30
Stefan	45	44	44	40	39	38	35	30

in this sequence, which contains a zoom out and a slow left pan, and the specific BPSI capabilities to handle complex camera motions. In addition, this sequence also has high contrast, which benefits the search for the optimal perspective deformation.

- For the Hall Monitor sequence in QCIF, gains up to 0.4 dB are obtained in terms of SI PSNR quality. These gains can be explained by the complex object motion associated to the two persons walking in the corridor. For the Bus QCIF sequence, gains up to 0.66 dB are obtained, while the Stefan QCIF sequence shows gains up to 0.37 dB. For the Stefan sequence, the BPSI framework is not able to estimate the motion model parameters as efficiently as for some other sequences since high camera motion occurs. Thus, considering a 15-Hz frame rate (where the key frames are less correlated) and the rather high motion, lower SI quality gains are obtained when compared to the results obtained for the other video sequences.
- For the Common Intermediate Format (CIF) spatial resolution with a 30-Hz frame rate, the SI quality gains are comparable to the SI quality QCIF results. For the Mobile and Calendar sequence, gains up to 0.89 dB are achieved, while the Bus sequence obtains 0.82 dB, the Stefan sequence 0.26 dB, and the Hall Monitor sequence 0.19 dB. For the sequences with high camera motion (such as Mobile and Calendar, and Bus sequences) and complex deformations (such as the Stefan sequence), the perspective motion model can better characterize the true motion when compared to the translational model.

Table 3 DVC-BPSI SI PSNR quality gains for QCIF@15Hz video sequences

Video sequence	Q_i	MCFI [dB]	BPSI [dB]	Δ_{BPSI} [dB]
Hall Monitor	Q_2	31.35	31.54	0.19
	Q_7	34.51	34.92	0.40
Mobile and Calendar	Q_2	26.21	26.93	0.72
	Q_7	28.06	29.06	*1.00*
Bus	Q_2	21.90	22.32	0.42
	Q_7	23.92	24.58	0.66
Stefan	Q_2	20.51	20.67	0.15
	Q_7	22.16	22.53	0.37

Table 4 DVC-BPSI SI PSNR quality gains for CIF@30Hz video sequences

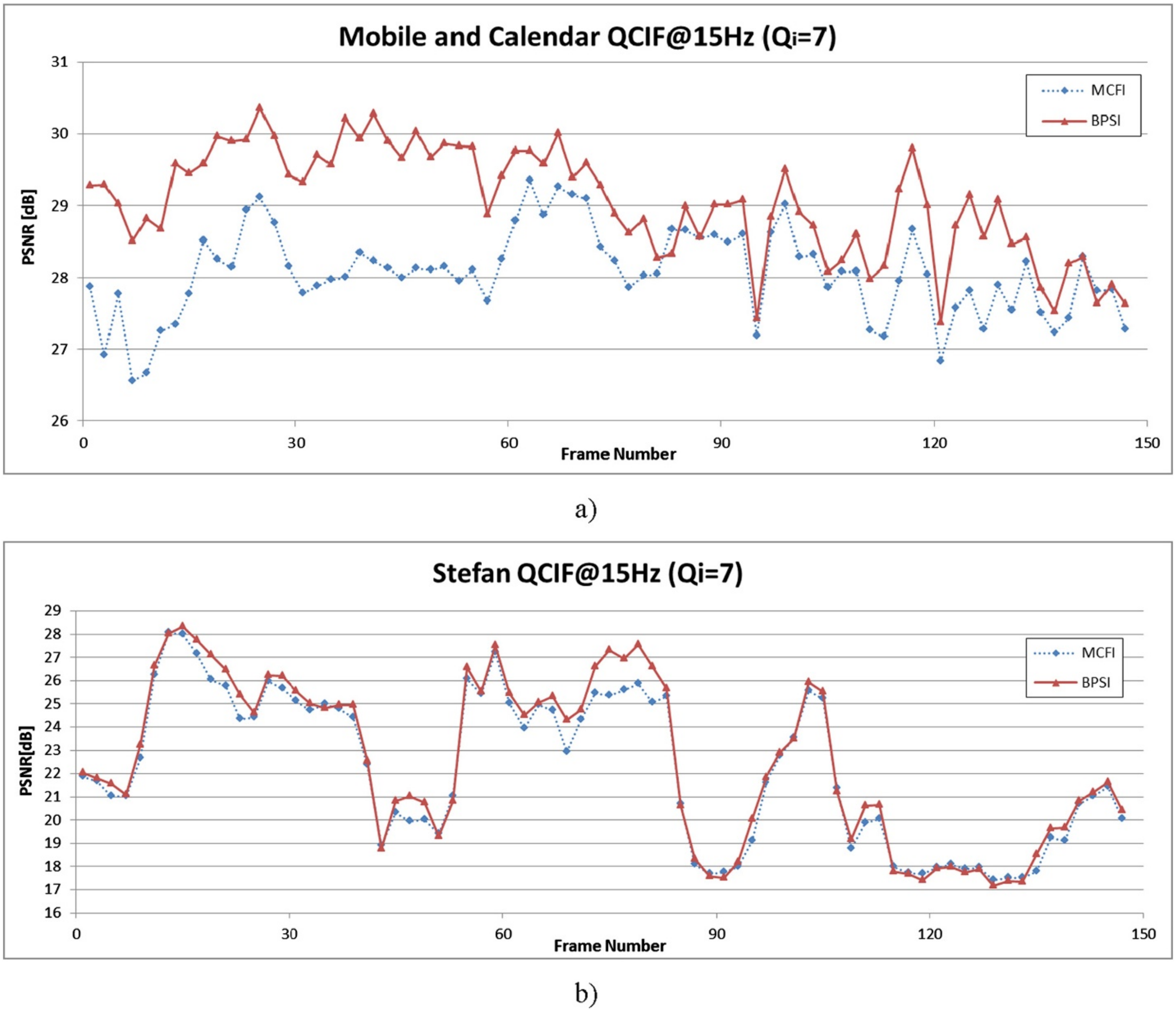
Video sequence	Q_i	MCFI [dB]	BPSI [dB]	Δ_{BPSI} [dB]
Hall Monitor	Q_2	33.07	33.16	0.09
	Q_7	35.64	35.83	0.19
Mobile and Calendar	Q_2	26.50	27.26	0.76
	Q_7	28.19	29.09	*0.89*
Bus	Q_2	22.18	22.65	0.47
	Q_7	24.31	25.14	0.82
Stefan	Q_2	22.13	22.31	0.18
	Q_7	24.70	24.95	0.26

In Figure 13, the SI quality temporal evolution is shown for the Mobile and Calendar, and Stefan sequences. The quantization parameters adopted for the H.264/AVC key frame codec in this experiment correspond to the RD point Q_7. The SI quality results for the Mobile and Calendar sequence (see Figure 13a) show average gains around 1 dB for the BPSI solution regarding the MCFI solution. This sequence includes a zoom out from the beginning until frame 74 and then a slow camera panning until the end of the sequence. As shown in Figure 13a, gains up to 1.8 dB are obtained when the zoom out camera motion occurs, i.e., when the translational model cannot accurately capture the camera motion. From the SI quality results for the Stefan sequence (Figure 13b), it is also possible to conclude that significant BPSI gains are obtained for some parts of the sequence (regarding the MCFI solution), notably when more complex motions occur. These gains are coherent with the overall gains presented in Table 3.

Figure 14 shows a couple of examples with the SI created for regions in two frames of the Hall Monitor sequence using the proposed BPSI and the benchmark MCFI techniques; for reference, also the corresponding original frame regions are included. As shown, the proposed method can lead to relevant perceptual gains in a rather important object of the video sequence; in this case, the MCFI estimation leads to ghosting artifacts which are much improved using the proposed BPSI technique.

5.4 RD performance evaluation

The RD performance for the proposed DVC-BPSI solution and the adopted benchmarks is presented in Figure 15, for the eight defined RD points according to the adopted test conditions. Moreover, Table 5 shows the DVC-BPSI RD performance gains over DVC-MCFI using the Bitrate Bjontegaard delta (BD-Rate) [28] and BD-PSNR metrics for four quantization parameter sets corresponding to the



a)

b)

Figure 13 SI PSNR temporal evolution for the sequences in QCIF: (a) Mobile and Calendar and (b) Stefan.

Q_1, Q_4, Q_7, and Q_8 RD points. The Bjontegaard metrics enable the comparison of RD curves in terms of the average PSNR improvement or the average bitrate savings (positive values mean gains).

From the RD performance results obtained, the following conclusions may be drawn:

- *DVC-BPSI vs. DVC-MCFI* - For the video sequences adopted, the proposed DVC-BPSI codec always outperforms the state-of-the-art DVC-MCFI codec. In fact, the DVC-BPSI codec shows gains up to 0.66 dB (maximum) and 0.49 dB (in terms of BD-PSNR) in comparison with DVC-MCFI for the Mobile and Calendar video sequence. Even for the Hall Monitor low-motion video sequence, where no significant deformations are available, some RD gains were obtained, notably 0.12 dB in terms of BD-PSNR.

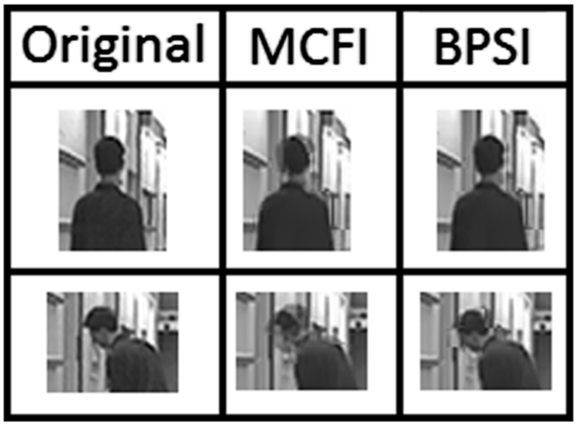
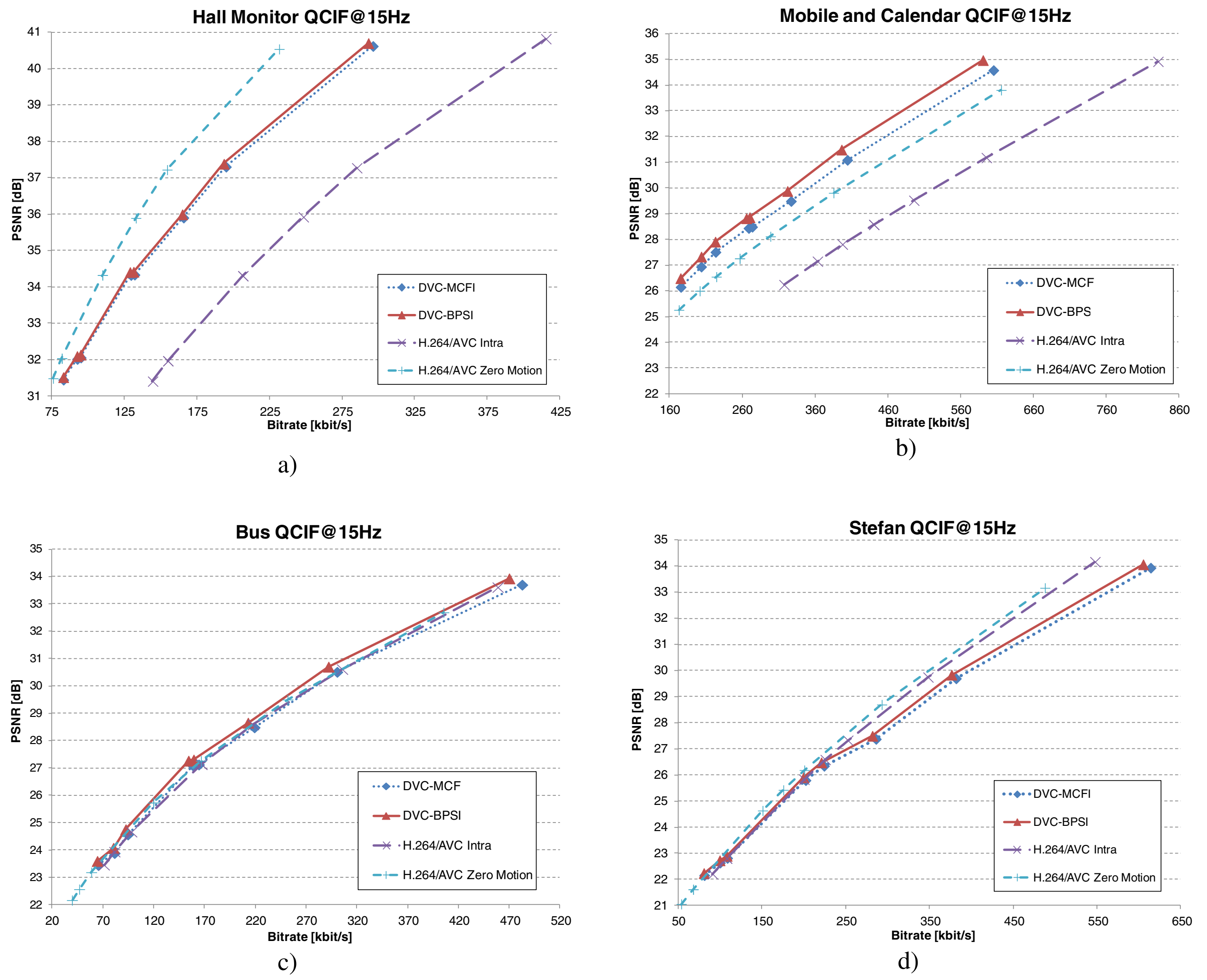


Figure 14 Hall Monitor SI creation example with the MCFI solution [9] and the proposed BPSI SI creation method. Frame nos. 23 (top) and 49 (bottom).



a)

b)

c)

d)

Figure 15 RD performance comparison for the sequences: (a) Hall Monitor, (b) Mobile and Calendar, (c) Bus, and (d) Stefan.

The perspective motion model used for the BPSI creation process is the main responsible for the RD performance gains since all the remaining techniques are kept the same. Thus, these RD performance results in the context of DVC codecs validate the assumption that perspective motion models can more accurately represent the motion in a video sequence. Generally, video sequences with complex camera motion (such as Mobile and Calendar, and Bus) can be better characterized with more complex motion models, and thus, higher RD performance gains can be obtained.

- *DVC-BPSI vs. H.264/AVC Intra* - As observed, the DVC-BPSI codec is able to outperform the H.264/AVC Intra codec for all video sequences but for the Stefan sequence; in fact, BD-PSNR gains up to 3.75 dB (see Table 5) are obtained for the Mobile and Calendar sequence; a similar behavior can be observed for the Hall Monitor video sequence with gains up to 3.56 dB. The complex and high motion

Table 5 Bjoontegard metric RD performance

Video sequence	DVC-BPSI vs. DVC-MCFI		DVC-BPSI vs. H.264/AVC Intra		DVC-BPSI vs. H.264/AVC Zero-Motion	
	BD-Rate [%]	BD-PSNR [dB]	BD-Rate [%]	BD-PSNR [dB]	BD-Rate [%]	BD-PSNR [dB]
Hall Monitor	−1.72	0.124	−55.36	3.56	+15.9	−1.31
Bus	−6.36	0.321	−8.85	0.449	−6.04	0.304
Mobile and Calendar	*−7.48*	*0.494*	*−60.58*	*3.75*	*−22.88*	*1.41*
Stefan	−3.30	0.187	+2.00	−0.104	+6.56	0.335

in the Stefan sequence leads to a significant amount of 'estimation errors' in the SI frame which causes lower RD performance. In such cases, better RD performance may be obtained with learning DVC solutions that are able to exploit already decoded data or hint DVC solutions that are able to exploit auxiliary information coded and transmitted by the encoder, e.g., block hashes.

- *DVC-BPSI vs. H.264/AVC Zero Motion* - The DVC-BPSI codec shows BD-PSNR gains up to 0.3 dB for the Bus sequence and up to 1.4 dB for the Mobile and Calendar sequence regarding the H.264/AVC Zero Motion codec. The impressive Mobile and Calendar RD performance gains can be explained by the fact that the DVC-BPSI codec can properly estimate (at the decoder) the slow panning (and zooming) while the H.264/AVC Zero Motion codec cannot since no motion estimation is performed at all. However, the DVC-BPSI codec does not outperform the H.264/AVC Zero Motion codec for the Hall Monitor and Stefan sequences with losses up to 1.3 dB. For the Hall Monitor sequence, the H.264/AVC Zero Motion video codec is able to efficiently characterize the static areas with the Skip mode (where no residual or MV data are transmitted), while for the Stefan sequence, the BPSI framework is still unable to create SI with competitive quality. However, the H.264/AVC Zero Motion codec has higher encoding complexity regarding the DVC-BPSI codec, as shown in the encoding complexity evaluation presented in [25].

Finally, it is important to stress that the proposed DVC-BPSI encoding complexity is kept low since no changes were made at the encoder side. The proposed BPSI framework can also be used for other objectives, such as error concealment in the context of a predictive video decoder, notably when an entire frame is lost (which happens frequently in packet loss networks), and in frame rate-up conversion scenarios when a low frame rate video is transmitted but a higher display rate is desired.

6. Conclusions

The main objective of this paper was to improve the overall RD performance of a state-of-the-art DVC codec using a more powerful SI creation framework, notably exploiting perspective transform motion modeling to better characterize more complex video motions. With the proposed DVC-BPSI codec, gains up to 1 dB were obtained in terms of SI quality and up to 0.5 dB in terms of RD performance when compared to the previous state-of-the-art MCFI and DVC-MCFI solutions, respectively. Regarding H.264/AVC Intra, RD gains of up to 4 dB are obtained, for low- and medium-motion sequences; when compared to the H.264/AVC Zero Motion, RD performance gains up to 1.5 dB can be obtained for sequences with strong global camera motion. Future work shall consider the development of an algorithm to merge neighboring perspective transforms according to some similarity criteria. In this way, it should be possible to eliminate some wrongly chosen deformations, thus providing a more coherent set of perspective transforms.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) through the project Multiview Video Disparity Compensation using Geometric Transforms (MuViDisCo) with reference IT/LA/P01082/2011.

Author details

[1]Instituto Superior Técnico – Instituto de Telecomunicações, Lisbon 1049-001, Portugal. [2]Instituto Superior de Engenharia de Lisboa – Instituto de Telecomunicações, Lisbon 1049-001, Portugal.

Received: 25 June 2013 Accepted: 20 November 2013
Published: 23 December 2013

References

1. T Wiegand, GJ Sullivan, G Bjøontegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Sys. Video Technol **13**(7), 560–576 (2003)
2. GJ Sullivan, J Ohm, H Woo-Jin, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Sys. Video Technol. **22**(12), 1649–1668 (2012)
3. J Ohm, GJ Sullivan, H Schwarz, TK Tan, T Wiegand, Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). IEEE Trans. Circuits Sys. Video Technol. **22**(12), 1669–1684 (2012)
4. F Bossen, B Bross, K Suhring, D Flynn, HEVC complexity and implementation analysis. IEEE Trans. Circuits Sys. Video Technol. **22**(12), 1685–1696 (2012)
5. D Slepian, J Wolf, Noiseless coding of correlated information sources. IEEE Trans. Info. Theory **19**(4), 471–480 (1973)
6. J Ziv, A Wyner, The rate-distortion function for source coding with side information at the decoder. IEEE Trans. Info. Theory **22**(1), 1–10 (1976)
7. F Pereira, L Torres, C Guillemot, T Ebrahimi, R Leonardi, S Klomp, Distributed video coding: selecting the most promising application scenarios. Signal Processing Image Comm. **23**(5), 339–352 (2008)
8. X Artigas, J Ascenso, M Dalai, S Klomp, D Kubasov, M Ouaret, The DISCOVER codec: architecture, techniques and evaluation, in *Picture Coding Symposium*. Lisbon, Portugal, 7–9 Nov 2007
9. J Ascenso, C Brites, F Pereira, Content adaptive Wyner-Ziv video coding driven by motion activity, in *IEEE International Conference on Image Processing*. Atlanta, USA, 16–19 Sept 2007
10. B Girod, A Aaron, S Rane, D Rebollo-Monedero, Distributed video coding. Proc. IEEE **93**(1), 71–83 (2005)
11. D Kubasov, C Guillemot, Mesh-based motion-compensated interpolation for side information extraction in distributed video coding, in *IEEE International Conference on Image Processing*. Atlanta, USA, 8–11 Oct 2006
12. M Tagliasacchi, S Tubaro, A Sarti, On the modeling of motion in Wyner-Ziv video coding, in *IEEE International Conference on Image Processing*. Atlanta, USA, 8–11 Oct 2006
13. G Petrazzuoli, M Cagnazzo, B Pesquet-Popescu, High order motion interpolation for side information improvement in DVC, in *IEEE International Conference on Acoustics Speech and Signal Processing*. Dallas, USA, 15–19 Mar 2010
14. Y Zhang, D Zhao, H Liu, Y Li, S Ma, W Gao, Side information generation with auto regressive model for low-delay distributed video coding. J. Visual Comm. Image Represent. **23**(1), 229–236 (2012)
15. X Huang, S Forchhammer, Improved side information generation for distributed video coding, in *IEEE International Workshop on Multimedia Signal Processing*. Cairns, Australia, 8–10 Oct 2008

16. X Huang, LL Raket, HV Luong, M Nielsen, F Lauze, S Forchhammer, Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow, in *IEEE International Workshop on Multimedia Signal Processing*. Hangzhou, China, 17–19 Oct 2011
17. M Cagnazzo, T Maugey, B Pesquet-Popescu, A differential motion estimation method for image interpolation in distributed video coding, in *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Taipei, Taiwan, 19–24 Apr 2009
18. A Abou-El Ailah, F Dufaux, J Farah, M Cagnazzo, B Pesquet-Popescu, Fusion of global and local motion estimation for distributed video coding. IEEE Trans. Circuits Sys. Video Technol **23**(1), 158–172 (2013)
19. WK Pratt, *Digital Image Processing*, 4th edn. (Wiley Interscience, New York, 2007)
20. S Faria, *Very low bit rate video coding using geometric transform motion compensation*. PhD thesis (University of Essex, 1996)
21. J Sung, S Park, J Park, B Jeon, Picture-level parametric motion representation for efficient motion compensation, in *IEEE International Conference on Image Processing*. Brussels, Belgium, 11–14 Sept 2011
22. A Glantz, M Tok, A Krutz, T Sikora, A block-adaptive skip mode for inter prediction based on parametric motion models, in *IEEE International Conference on Image Processing*. Brussels, 11–14 Sept 2011
23. A Glantz, A Krutz, T Sikora, Adaptive global motion temporal prediction for video coding, in *Picture Coding Symposium*. Nagoya, Japan, 7–10 Dec 2010
24. J Ascenso, C Brites, F Pereira, Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding, in *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*. Smolenice, Slovak Republic, 29 June–2 July 2005
25. J Ascenso, Improving compression efficiency in distributed video coding systems, PhD dissertation, Instituto Superior Técnico, Universidade Técnica de Lisboa, November 2010. http://www.img.lx.it.pt/publications_ig.html. Accessed 19 Dec 2013
26. C Brites, F Pereira, Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding. IEEE Trans. Circuits Sys. Video Technol. **18**(9), 1177–1190 (2008)
27. J Ascenso, C Brites, F Pereira, Design and performance of a novel low-density parity-check code for distributed video coding, in *IEEE International Conference on Image Processing*. San Diego, USA, 12–15 Oct 2008, 2008
28. G Bjontegaard, Calculation of average PSNR differences between RD curves, in *13th ITU-T VCEG Meeting*. Austin, 2–4 Apr 2001

doi:10.1186/1687-6180-2013-189
Cite this article as: Monteiro *et al.*: Perspective transform motion modeling for improved side information creation. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:189.