Multiview plus depth video coding with temporal prediction view synthesis

Andrei I. Purica, Elie G. Mora, Beatrice Pesquet-Popescu, *Fellow, IEEE*, Marco Cagnazzo, *Senior Member, IEEE* and Bogdan Ionescu, *Senior Member, IEEE*

***Abstract*—Multiview video plus depths formats use view synthesis to build intermediate views from existing adjacent views at the receiver side. Traditional view synthesis exploits the disparity information to interpolate an intermediate view by taking into account inter-view correlations. However, temporal correlation between different frames of the intermediate view can be used to improve the synthesis. We propose a new coding scheme for 3D-HEVC that allows us to take full advantage of temporal correlations in the intermediate view and improve the existing synthesis from adjacent views. We use optical flow techniques to derive dense motion vector fields from the adjacent views and then warp them at the level of the intermediate view. This allows us to construct multiple temporal predictions of the synthesized frame. A second contribution is an adaptive fusion method that judiciously selects between temporal and inter-view prediction in order to eliminate artifacts associated with each prediction type. The proposed system is compared against the state-of-the-art VSRS-1DFast technique used in 3D-HEVC standardization. 3 intermediary views are synthesized. Gains of up to 1.21 dB Bjontegaard Delta PSNR are shown, when evaluated on several standard multiview video test sequences.**

***Index Terms*—multiview video plus depth, 3DV, temporal and inter-view prediction, view synthesis, 3D-HEVC.**

I. Introduction

RECENT advances in video acquisition, compression and transmission technologies have brought significant market potential for immersive communications. Common examples [1] [2] include immersive teleconference systems, 3D video, holography and Free Viewpoint Television (FTV). A typical format for some of these applications is the MultiView Video (MVV) composed of a set of N video sequences representing the same scene, referred to as views, acquired simultaneously by a system of N cameras positioned under different spatial configurations. An alternative representation is the Multiview-Video-Plus-Depth format (MVD) [3], where the depth information is used in addition to texture for each viewpoint. This allows for a less costly synthesis of much more virtual views, using for example Depth-Image-Based-Rendering (DIBR) methods [4].

View synthesis is the process of extrapolating or interpolating a view from other available views. It is a popular research topic in computer vision, and numerous methods have been developed in this field over the past four decades. View synthesis techniques can be mainly classified in three categories [5]. The methods in the first category, like DIBR, require explicit geometry information such as depth or disparity maps to warp the pixels in the available views to the correct position in the synthesized view [6] [7]. Methods in the second category require only implicit geometry, like some pixel correspondences in the available and synthesized view, that can be computed using optical flow [8] [9] for instance. Finally, methods in the third category require no geometry at all. They appropriately filter and interpolate a pre-acquired set of samples (examples of tools in this category include light field rendering [10], lumigraph [11], concentric mosaics [12]). A common problem in view synthesis are areas that are occluded in the available views but should be visible in the virtual ones. These areas appear as holes in virtual views, also referred to as disocclusions . This problem is currently resolved by using inpainting algorithms such as the ones described in [13] and [14]. Two of the most popular inpainting algorithms were developed by Bertalmio and Sapiro [15] and Criminisi *et al.* [16].

Recently, the Moving Pictures Experts Group (MPEG) expressed a significant interest in MVD formats for their ability to support 3D video applications. This new activity is mainly focused on developing a 3D extension of the HEVC [17] video coding standard, after a first standardization activity finalized with Multiview Video Coding (MVC) [18]. An experimental framework was developed as well, in order to conduct the evaluation experiments [19]. This framework defined a View Synthesis Reference Software (VSRS) as part of the 3D-HEVC test model [20], which would later become an anchor to several new rendering techniques. Furthermore, establishing whether encoding all views or synthesizing some from coded views is better for multiview video sequences is still an open matter. Recently MPEG decided to dedicate 6 months to compare the two schemes [21].

Traditionally, view synthesis methods, and VSRS in particular, only use inter-view correlations to render virtual views. However temporal correlations can also be exploited to improve the quality of the synthesis. In general, this type of methods synthesize or improve the synthesis of a frame by extracting additional information from different time instants, as opposed to DIBR methods which only use adjacent views at the same time instant. For instance in [22] the authors use motion vector fields between frames of the intermediate views to improve the view synthesis in MVC standard. Chen *et al.* [23] use motion vector fields computed through block-based motion estimation in the reference views and then warp both the start and end point of the vectors in the synthesized view. The motion vectors are then used to retrieve information about dis-occluded regions from other frames. Sun *et al.* [24] and Kumar *et al.* [25] use adjacent views to extract background information from multiple time instants,

used for hole filling in a DIBR synthesis. In [26] the authors use the information from the current and other frames of the synthesized video to fill hole regions. Other studies use the inter-view correlations directly during coding, view-synthesis prediction (VSP) [27] [28] [29] or take advantage of multiview format redundancies to deal with network packet loss [30]. Yuan *et al.* [31] use Weiner filter to improve the synthesis by eliminating distortions caused by coding.

In this paper, we propose a new coding scheme for 3D-HEVC built around a novel view synthesis method that fully exploits temporal and inter-view correlations. Our method is designed to complement the synthesis method used in the 3D-HEVC standardization process in order to improve the quality of the synthesis. We use the optical flow to derive dense motion vector fields between frames in the adjacent views which are available at the decoder side, then warp them at the level of the intermediate view. This allows us to build different temporal predictions from left and right adjacent views using reference frames at two time instants (past and future). Other motion estimation techniques that are less computationally intensive can also be used at the cost of prediction accuracy [32] [33] [34]. However, since it does not require sending any residual information, we prefer using an optical flow motion estimation technique, since it offers a more accurate prediction [35]. The reference frames used for motion compensation are previously encoded and sent as an additional frame per GOP in the intermediate view, we will refer to these frames as key frames in the rest of this paper. The four predictions are then merged into a single one, with the aim of reducing the number of holes in the final synthesis. Due to a big temporal distance between reference and synthesized frames, the motion vector fields may be imprecise especially for frames with intense motion. We mitigate these effects by using the so-called "Hierarchical" synthesis scheme, in which temporal layers are used to perform symmetric synthesis (where each frame is synthesized from either a key frame or a previously synthesized frame) and we compare it with a "Direct" scheme (where each frame is directly synthesized from a past and a future key frame). To further improve the quality of the synthesis, we introduce an adaptive fusion method that selects between inter-view and temporal prediction. The remaining disocclusions in the synthesized image are then filled by a linear inpainting method.

The remaining of this paper is organized as follows. The second section of this paper presents a state-of-the-art of view synthesis techniques. The proposed method is described in the third section. The results obtained are summarized in Section IV with a detailed interpretation, and finally conclusions and future work directions are presented in Section V.

II. State of the art of view synthesis techniques

In this state of the art, we focus on the first class of view synthesis methods, also referred to as DIBR techniques. We first discuss the rendering technique used in the reference software for view synthesis, and in the rendering software used by the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [36]. Then, an overview of other rendering techniques found in the literature is presented.

A. Reference software

1) View Synthesis Reference Software: VSRS inputs two texture views and their two associated depth maps, along with intrinsic and extrinsic camera parameters. The output is a synthesized intermediate view. VSRS allows synthesizing frames using two operational modes: a general mode and a 1D mode, respectively used for non-parallel (e.g. cameras aligned in an arc) and 1D-parallel (cameras are aligned in a straight line perpendicularly to their optical axes) camera settings.

Figure 1 illustrates the rendering process in the general mode of VSRS. First, the left and right reference depth maps ($s_{D,l}$ and $s_{D,r}$) are warped to the virtual view position, giving $s'_{D,l}$ and $s'_{D,r}$. The occlusions are handled by the highest depth value (closest to the camera), usually the depth values are reversed quantified from 0 to 255 such that the highest value in the depth map corresponds to the lowest depth of the scene [1]. $s'_{D,l}$ and $s'_{D,r}$ are then median filtered to fill small holes, giving $s''_{D,l}$ and $s''_{D,r}$. A binary mask is maintained for each view to keep track of larger holes caused by disocclusions. $s''_{D,l}$ and $s''_{D,r}$ are then used to warp the texture views $s_{T,l}$ and $s_{T,r}$ to the virtual view position, giving $s'_{T,l}$ and $s'_{T,r}$ (this reverse warping process wherein the depths are warped first and then used to warp the texture is reported to give a higher rendering quality [19]). Holes in one of the warped views are filled with collocated non-hole pixels from the other warped view, if available. This gives $s''_{T,l}$ and $s''_{T,r}$, which are then blended together to form a single representation. The blending can be a weighted average according to the distance of each view to the virtual view point (Blending-On mode), or it can simply consist in taking the closest view to the virtual view point, and discarding the other (Blending-Off mode). The binary masks of each view are merged together at this stage and the remaining holes are filled at the final stage of the algorithm by propagating the color information inward from the region boundaries.

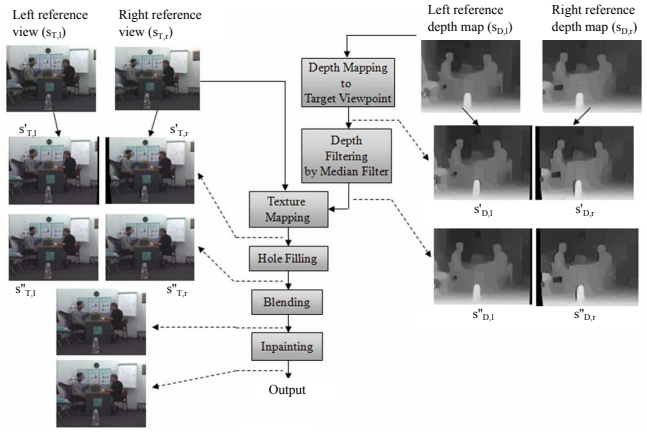


Fig. 1. Flow diagram for View Synthesis Reference Software (VSRS) general mode [19].

The 1D mode of VSRS works a bit differently. In this mode, the camera setup is assumed to be 1D parallel. This allows to make a number of simplifications to the warping process which is reduced to a simple horizontal shift. First, the color video is up-sampled for half-pixel or quarter pixel accuracy.

A "CleanNoiseOption" and "WarpEnhancementOption" avoid warping unreliable pixels. The process gives two warped images, two warped depth maps and two binary masks from the left and right reference views. Each pair is then merged together. When a pixel gets mapped from both the left and the right reference views, the final pixel value is either the pixel closest to the camera or an average of the two. Remaining holes are filled by propagating the background pixels into the holes along the horizontal row. Finally, the image is downsampled to its original size.

2) View Synthesis Reference Software 1D Fast: Each contribution to the 3D-HEVC standardization that proposes to modify the coding of dependent views or depth data, is required to present coding results on synthesized views. The software used for synthesizing the intermediate views is a variant of VSRS, called View Synthesis Reference Software 1D Fast (VSRS-1DFast). This software is included in the HTM package, and is documented in the 3D-HEVC test model [20]. VSRS-1DFast allows inputting two or three texture and depth views along with their corresponding camera parameters, and synthesize an arbitrary number of intermediate views. Just like the 1D mode of VSRS, VSRS-1DFast assumes that the camera setup is 1D parallel. Figure 2 illustrates the different steps of

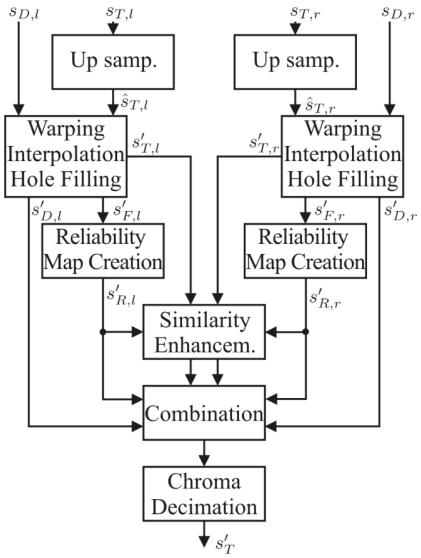


Fig. 2. Flow diagram for View Synthesis Reference Software 1D Fast (VSRS-1DFast) [20].

the rendering algorithm used in VSRS-1DFast. The texture views $s_{T,l}$ and $s_{T,r}$ are first upsampled to obtain $\widehat{s}_{T,l}$ and $\widehat{s}_{T,r}$: the luma component is upsampled by a factor of four in the horizontal direction, and the chroma by a factor of eight in horizontal direction and two in vertical direction, thus yielding the same resolution for all components. The warping, interpolation and hole filling are carried out for $\widehat{s}_{T,l}$ and $\widehat{s}_{T,r}$ line-wise. This gives two representations of the synthesized frame: $s'_{T,l}$ and $s'_{T,r}$. Then, two reliability maps $s'_{R,l}$ and $s'_{R,r}$ are determined indicating which pixels correspond to disocclusions (reliability of 0). A similarity enhancement stage then adapts the histogram of $s'_{T,l}$ to the one of $s'_{T,r}$. Finally, $s'_{T,l}$ and $s'_{T,r}$ are combined. If the "interpolative rendering" option is activated, the combination would depend on the warped depth maps and the two reliability maps created. If not, the synthesized view is mainly rendered from one view and only the holes are filled from the other view. The resulting combination is later down-sampled to the original size of the texture views.

B. Rendering techniques in literature

In [37], a rendering technique called View Synthesis using Inverse Mapping (VSIM) is introduced. It operates at full-pel accuracy and assumes a 1D-parallel camera setting. The left and right texture views are warped to the synthesized view position using simple horizontal shifts, also called column shifts. A table is maintained for the left and right interpretations of the synthesized frame which records the column shift of each pixel. Holes in these two tables are filled using a median filter. Then, the two representations are merged and the remaining holes are filled by checking the collocated value in the tables, and inverse mapping the pixel back to its original value in the left or right view. Residual holes are filled by simply assuming that their depth is the same as the depth of the collocated pixels in the original views. VSIM outperforms VSRS, on average, by 0.41 dB at quarter-pel accuracy and by 1.35 dB at full-pel accuracy on 5 sequences. However, the rendering runtime is not provided, making it difficult to assess the complexity of the method.

In [38], the depth maps are pre-processed with an adaptive smoothing filter in order to reduce holes after synthesis. The filter is only applied to edges in the depth map (corresponding to an abrupt transition in depth values) since these are the main cause for holes. The method is thus less complex than methods which apply a symmetric or asymmetric smoothing filter to the entire depth map. Furthermore, if hole regions correspond to vertical edges, an asymmetric Gaussian smoothing filter is used to further pre-process the depth map. No objective gains are reported, but a perceptual improvement is noticed on some synthesized sequences.

A technique that does not require pre-processing the depth map is introduced in [39]. A hole in the synthesized texture image is filled by the color of the neighboring pixel (between the 8 direct neighboring pixels) with the smallest depth value in the synthesized depth map (this is referred to as Horizontal, Vertical and Diagonal Extrapolation (HVDE)). The two warped texture images are complemented (holes in one are filled with available pixel values in the other), and later blended, giving a final image W. The same process (HVDE, complementation, and blending) can also be performed in case the depth maps were pre-processed with a bi-lateral smoothing filter, giving an image A, which would then be used to fill remaining holes in W. This technique is reported to outperform basic DIBR by 1.78 dB on one sequence.

Another method for improving the quality of the synthesis is to apply a non-linear transformation to the depth maps [40]. Specifically, the depth range of points in the background is compressed, such that these points would have the same or slightly different depths. This reportedly reduces holes in the synthesis. The transformation depends on the depth map

histogram. Objective gains are not presented but a visible improvement is noticed on the shown images.

Another desired feature is the possibility to freely change the quality of a synthesized view. Since the quality of DIBR rendering depends on the actual synthesis process, additional boundary artifact processing can be used to adjust the quality of the synthesis. Zhao *et al.* analyze and reduce the boundary artifacts from a texture-depth alignment perspective in [41]. In [42] Cheung *et al.* tackle the problem of bit allocation for DIBR multiview coding. The authors use a cubic distortion model based on DIBR properties and demonstrate that the optimal selection of QPs for texture and depth maps is equivalent to the shortest path in a specially constructed 3D trellis. Xiao *et al.* [43] propose a scalable bit allocation scheme, where a single ordering of depth and texture packets is derived. Furthermore, depth packets are ordered based on their contribution to the reduction of the synthesized view distortion.

Other works also exploit pixel-based processing with dense MVFs with an end goal of improving the synthesis at the decoder side. Li *et al.* compute dense MVFs on texture in [44]. Time consuming optical flow computations are limited only around the edges of objects. Additional depth predictors are obtained by mapping the MVs computed on texture to depth. The depth map improvement is reflected in a high increase of quality for synthesized views.

C. Remarks

The rendering techniques used in the reference softwares, and in most contributions in literature, are all based on 3D image warping using depth maps. Pixels from reference views are mapped to pixels in the virtual view using the disparity information that the depth maps convey. However, we show that the synthesis can be improved by extending DIBR to the temporal axis. In the remaining work, we present a rendering method where temporal correlations between different frames in the synthesized views are exploited to improve the quality of the synthesis. Our method is detailed in the next section.

III. Proposed method

Traditional rendering techniques synthesize an intermediate frame only from the left and right reference views at the same time instant. By exploiting the temporal correlations in the multiview sequence, we are able to obtain additional predictions from past and future frames and merge them together to obtain the synthesized frame. We refer to our synthesis method as View Synthesis exploiting Temporal Prediction (VSTP). In this section, we describe the epipolar constraint for disparity maps and optical flows, on which the proposed method is based. We then provide a description of the algorithm and propose two synthesis schemes for a Group Of Pictures (GOP) that exploit this idea.

A. Epipolar constraint

Figure 3 shows the relation between the positions of a real-world point projection in different views and at different time instants. Let us consider I^r_{t-1}, I^r_t, I^s_{t-1}, I^s_t which are, respectively, the reference (r) view frames and the synthesized (s) view frames at time instants $t-1$ and t. Let $M \times N$ be the size of the image with M being the height and N the width. Let $\mathbf{k} = (x, y)$ be a point in I^r_{t-1}, $\mathbf{v}_r(\mathbf{k})$ its associated motion vector (I^r_t is the reference frame for I^r_{t-1}), pointing to a corresponding point in I^r_t, and $\mathbf{d}_{t-1}(\mathbf{k})$ its associated disparity vector, pointing to a corresponding point in I^s_{t-1}. Let $\mathbf{v}_s(\mathbf{k} + \mathbf{d}_{t-1}(\mathbf{k}))$ be the motion vector of the projection of $\mathbf{k}$ in I^s_{t-1} and $\mathbf{d}_t(\mathbf{k} + \mathbf{v}_r(\mathbf{k}))$ the disparity vector of the projection of $\mathbf{k}$ in I^r_t. If the point is not occluded, there is only one projection of $\mathbf{k}$ in I^s_t, so the two vectors will point to the same position. This defines a so-called epipolar constraint [45] on $\mathbf{k}$, which can be written as:

$$\mathbf{v}_r(\mathbf{k}) + \mathbf{d}_t(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) = \mathbf{d}_{t-1}(\mathbf{k}) + \mathbf{v}_s(\mathbf{k} + \mathbf{d}_{t-1}(\mathbf{k})) \quad (1)$$

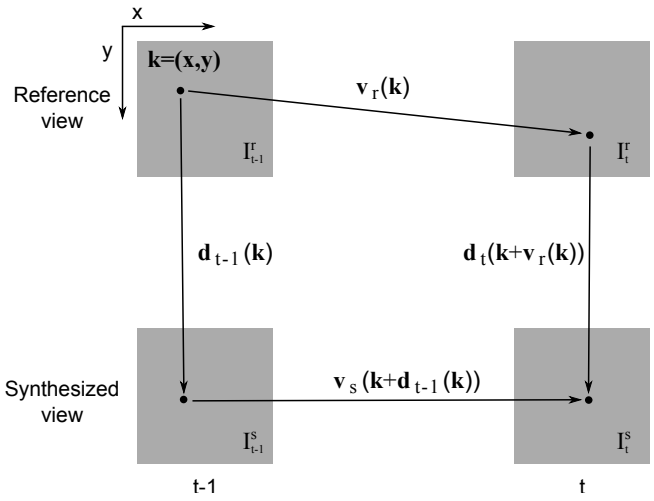


Fig. 3. Epipolar constraint, the relation between the disparity fields $\mathbf{d}_{t-1}$ and $\mathbf{d}_t$ at two time instants t and $t-1$ respectively, and the motion vector fields in the synthesized and reference view $\mathbf{v}_s$ and $\mathbf{v}_r$ respectively for a position $\mathbf{k}$ in the reference frame I^r_{t-1}.

B. Method description

The goal of the method is to synthesize I^s_t from a past and future key frame in the synthesized view. Knowing $\mathbf{v}_r$, $\mathbf{d}_t$, and $\mathbf{d}_{t-1}$, $\mathbf{v}_s$ can be derived using Equation (1) for every pixel in I^s_{t-1} that has a correspondence in I^r_{t-1}:

$$\mathbf{v}_s(\mathbf{k} + \mathbf{d}_{t-1}(\mathbf{k})) = \mathbf{v}_r(\mathbf{k}) + \mathbf{d}_t(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) - \mathbf{d}_{t-1}(\mathbf{k}) \quad (2)$$

$\mathbf{v}_r$ can be obtained by inputting I^r_{t-1} and I^r_t in an optical flow algorithm [46]. The result is a dense motion vector field $\mathbf{v}_r$ where each pixel in I^r_{t-1} is associated with a motion vector. The disparity maps $\mathbf{d}_t$ and $\mathbf{d}_{t-1}$ can be obtained by simply converting the values in the depth maps Z^r_t and Z^r_{t-1} associated with I^r_t and I^r_{t-1} respectively into disparity values. We assume that we are dealing with a 1D parallel camera setup, and that only horizontal disparities exist. In this simple setup, the disparity value for a point $\mathbf{k}$ of coordinates (x, y)

in I^r_{t-1} can be written as:

$$\mathbf{d}^x_t(\mathbf{k}) = f \cdot B \left[\frac{Z^r_t(x,y)}{255} \left(\frac{1}{Z_{min}} - \frac{1}{Z_{max}} \right) + \frac{1}{Z_{max}} \right] \quad (3)$$
$$\mathbf{d}^y_t(\mathbf{k}) = 0$$

where f is the focal length of the camera, B the baseline between the reference and synthesized views, and Z_{min} and Z_{max} the extremal depth values. The same formula can be applied to obtain $\mathbf{d}_{t-1}$.

If we decompose Equation (2) for the x and y components separately, we obtain:

$$\begin{aligned} &\mathbf{v}^x_s(x + \mathbf{d}^x_{t-1}(x,y), y) \\ &= \mathbf{v}^x_r(x,y) + \mathbf{d}^x_t(x + \mathbf{v}^x_r(x,y), y + \mathbf{v}^y_r(x,y)) - \mathbf{d}^x_{t-1}(x,y) \\ &\mathbf{v}^y_s(x + \mathbf{d}^x_{t-1}(x,y), y) = \mathbf{v}^y_r(x,y) \end{aligned} \quad (4)$$

There will be holes in $\mathbf{v}_s$ that coincide with disocclusions created when warping I^r_{t-1} with the $\mathbf{d}_{t-1}$ disparity vector field. If two or more positions in I^r_{t-1}, $\mathbf{k}_1$ and $\mathbf{k}_2$ for instance, are warped to the same position $\mathbf{k}_3$ in I^s_{t-1} (occlusion), the vector $\mathbf{v}_s(\mathbf{k}_3)$ retained is the one which corresponds to the pixel with the highest depth value, as shown in Equation (5): the motion vectors for occluded points of the scene are thus ignored.

$$\mathbf{v}_s(\mathbf{k}_3) = \begin{cases} \mathbf{v}_r(\mathbf{k}_1) + \mathbf{d}_t(\mathbf{k}_1 + \mathbf{v}_r(\mathbf{k}_1)) - \mathbf{d}_{t-1}(\mathbf{k}_1) \\ \qquad \text{if } Z^r_{t-1}(\mathbf{k}_1) > Z^r_{t-1}(\mathbf{k}_2) \\ \mathbf{v}_r(\mathbf{k}_2) + \mathbf{d}_t(\mathbf{k}_2 + \mathbf{v}_r(\mathbf{k}_2)) - \mathbf{d}_{t-1}(\mathbf{k}_2) \\ \qquad \text{otherwise} \end{cases} \quad (5)$$

Using the motion vector field $\mathbf{v}_s$ and I^s_{t-1}, a prediction of I^s_t can be made, although it will contain holes due to disoccluded areas in $\mathbf{v}_s$. A total of four predictions can be made by exploiting the epipolar constraint, one for each reference view (left and right, L and R) and at each time instant (past and future, p and f), they will be denoted by $\mathcal{P}^{(i)}(I^s_t)$ where $i \in \{0,1,2,3\}$. This is shown in Figure 4.

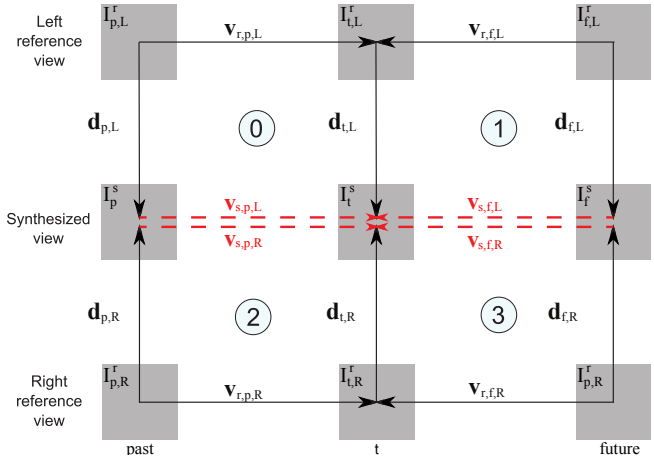


Fig. 4. Four predictions using the epipolar constraint: dotted lines represent the new temporal predictions introduced by our method.

The four predictions are then merged into a single one $\widetilde{I}^s_t$, where the value of each pixel equals the average of the non-disoccluded pixel values in the four predictions as shown in the following equation. When all four predictions contain the same disocclusion, the pixel value is computed by inpainting. Indeed, while the four predictions contain disocclusions, the majority of these holes are not the same in all predictions and thus they will be filled after the merging step:

$$\widetilde{I}^s_t(\mathbf{k}) = \begin{cases} \dfrac{\sum_{i=0}^{A(\mathbf{k})} \mathcal{P}^{(i)}(I^s_t(\mathbf{k}))}{A(\mathbf{k})} & \text{if } A(\mathbf{k}) \neq 0 \\ \text{inpainted} & \text{if } A(\mathbf{k}) = 0 \end{cases} \quad (6)$$

where $A(\mathbf{k})$ is the number of existing predictions for position $\mathbf{k}$. Disocclusions ($A(\mathbf{k}) = 0$) are filled using the same inpainting method used in VSRS-1DFast, which is a simple line-wise interpolation.

Figure 5 illustrates the steps of VSTP algorithm. In order to generate a temporal prediction, the algorithm inputs two frames of the reference view at two time instants, i.e., a current and a future or past time instants, denoted by $I^r_{t,L}$ and $I^r_{p,L}$ respectively in the figure, and computes a dense motion vector field between the two ($\mathbf{v}_{r,p,L}$). The dense MVF is then warped at the level of the synthesized view using the corresponding disparity maps ($\mathbf{d}_{t,L}$ and $\mathbf{d}_{p,L}$). We also retain a disparity map corresponding with the new MVF ($\mathbf{d}'$). Thus, each pixel has an associated motion vector and disparity. The next step is the backward motion compensation in which we use a key frame (I^s_p) as reference in order to obtain a first temporal prediction, in case of overlapping values we use $\mathbf{d}'$ to select the foreground pixel. $\widehat{I}^s_{p,R}$, $\widehat{I}^s_{f,L}$, $\widehat{I}^s_{f,R}$ are obtained using the same steps in the right reference view at the same time instant and at a future time instant in the left and right reference views respectively, as described in Figure 4. The final synthesis is obtained by performing a simple merge between the four temporal predictions or an inter-view/temporal fusion as described in Section III-D. The inter-view prediction is denoted by $\widehat{I}^i$ in Figure 5.

C. Prediction schemes in a GOP

The synthesized view is rendered GOP-wise in our algorithm. The GOP structure is the one used to code the left and right reference views. In addition to the reference views (as required by VSRS-1DFast) we send a first frame per GOP of the synthesized view (at the encoder side we require this view, it can be either original or synthesized from uncompressed adjacent views if not available) in the bitstream. These frames, referred to in the rest of this work as key frames, are efficiently coded using 3D-HEVC with the left view serving as inter-view reference (the base view). The rest of the frames are synthesized using our method with one of the temporal prediction schemes described below. For the first frame actually synthesized in a GOP, the key frame of the current GOP and the one of the future GOP respectively are the past and future reference frames, I^s_p and I^s_f respectively.

Figure 6 shows the difference between the two temporal prediction schemes. The "Direct" scheme uses the key frame of the current GOP and the one of the next GOP as past and future reference frames for all remaining frames to synthesize in the GOP. This results in an asymmetric prediction, with

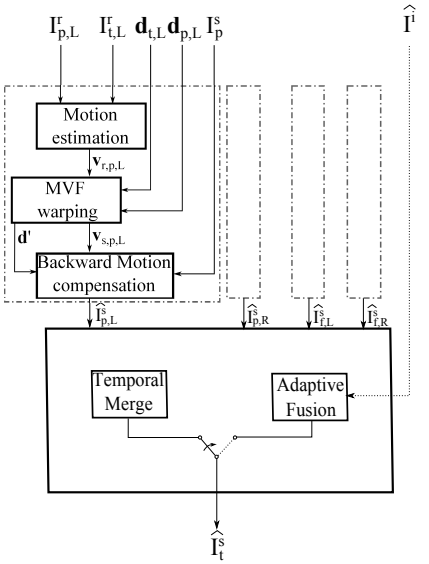
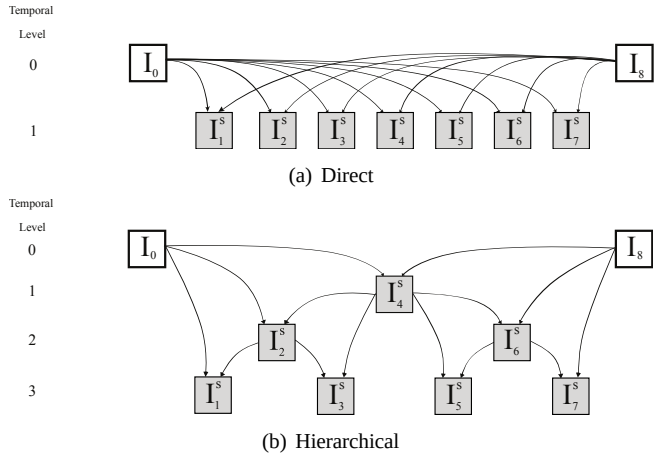


Fig. 5. Flow diagram for View Synthesis exploiting Temporal Prediction (VSTP). The dotted dash line is the temporal prediction block, which is applied four times, i.e. past and future time instants (p and f) in the left and right (L and R) reference (r) views.

two different temporal distances between each of the two key frames and the current frame. The temporal distance can be as high as the GOP size minus one, and an optical flow computation with such large temporal distances can give imprecise motion vector fields thus making the "Direct" scheme inefficient. An alternative scheme, called the "Hierarchical" scheme, can be used, in which temporal layers are used to perform symmetric predictions (with equal temporal distances). In each layer, the past and future references for the current frame are either the key frames or already synthesized frames in lower layers. The maximal temporal distance in this scheme equals half of the GOP size.



(a) Direct

(b) Hierarchical

Fig. 6. Temporal prediction schemes inside a GOP of the synthesized view.

D. Adaptive Fusion

In the proposed method the synthesized frame is obtained by merging our four temporal predictions as described in Equation (6). When dealing with fast moving objects, the optical flow computation between frames with high temporal distance may give imprecise motion vector fields which lead to an inconsistent positioning of the objects in the four temporal predictions. In this case, a simple average-based merging would result in a bad representation of objects with high motion intensity. In what follows, we refer to the traditional disparity based synthesis used in VSRS-1DFast as the inter-view prediction. We introduce a different merging algorithm called "Adaptive Fusion" which uses the inter-view prediction and our temporal prediction alternatively for different parts of the image. The idea of this method is to generate a binary fusion map in which we mark the bad pixels from the temporal prediction, to be replaced by the inter-view prediction. The first step of this algorithm is to estimate which areas will select the inter-view prediction and which ones will select the temporal prediction. The next step is the actual fusion, where each pixel value is computed as an average between either the temporal or inter-view predictions , depending on the previously computed binary map.

In order to describe our selection process for a pixel, let us consider: $\widehat{i}^t_{pL}$, $\widehat{i}^t_{fL}$, $\widehat{i}^t_{pR}$, $\widehat{i}^t_{fR}$ four temporal predictions of a pixel at position $\mathbf{k}$ and $\widehat{i}^i$ the blend between the left and right inter-view predictions obtained from VSRS-1DFast. It is safe to assume that good temporal predictions of a pixel are similar, i.e., the values are close to each other (have a low spread). On the contrary, imprecise motion vector fields might lead to dissimilar values that span over a large range (have a wide spread) and in this case inter-view prediction should be used. Note that in some cases $\widehat{i}^i$ is worse than the temporal prediction even if we have a wide spread. The challenge is to remove artifacts in the temporal prediction without introducing new ones from the inter-view prediction. By comparing the value of $\widehat{i}^i$ to our four temporal predictions we can identify four cases. In the following, the maximum and minimum value of the temporal predictions are denoted by $\widehat{i}^t_{max}$ and $\widehat{i}^t_{min}$ respectively :

Case 1: Wide spread and $\widehat{i}^i \in [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$
Case 2: Wide spread and $\widehat{i}^i \notin [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$
Case 3: Low spread and $\widehat{i}^i \in [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$
Case 4: Low spread and $\widehat{i}^i \notin [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$

We consider **Case 1** and **Case 4** as typical situations in which we should select inter-view and temporal predictions respectively. Indeed, in **Case 1**, wide spread means there is a bad match between the four temporal predicted values, which indicate an imprecise optical flow computation. An inter-view prediction inside this range is probably the best value. **Case 4** indicates a good temporal prediction and we should use the average of the four points. In **Case 2** the inter-view predicted value is either good or very bad depending on how far away it is from $\widehat{i}^t_{min}$ or $\widehat{i}^t_{max}$. In **Case 3** the two prediction values are close and we prioritize the temporal one. When dealing with disocclusions, the number of available temporal or inter-view predictions for a pixel can vary, i.e., a certain position (x, y) can be a disocclusion in one or more temporal or inter-view predictions. In situations when only one type of prediction is available we select it, and if we have no prediction at all, we mark the pixel to be later filled.

Considering the vectors $\mathbf{p_t} = [\widehat{i}^t_{pL}, \widehat{i}^t_{fL}, \widehat{i}^t_{pR}, \widehat{i}^t_{fR}]$ and $\mathbf{p_{t\&i}} = [\widehat{i}^t_{pL}, \widehat{i}^t_{fL}, \widehat{i}^t_{pR}, \widehat{i}^t_{fR}, \widehat{i}^i]$, the selection between inter-view and temporal prediction for a pixel is done as follows:

$$\widehat{i} = \begin{cases} \widehat{i}^t & \text{if } \mathrm{mean}(\mid \mathbf{p_t} - \mathrm{mean}(\mathbf{p_t}) \mid) \\ & -\mathrm{mean}(\mid \mathbf{p_{t\&i}} - \mathrm{mean}(\mathbf{p_{t\&i}}) \mid) < \alpha \\ \widehat{i}^i & \text{if } \mathrm{mean}(\mid \mathbf{p_t} - \mathrm{mean}(\mathbf{p_t}) \mid) \\ & -\mathrm{mean}(\mid \mathbf{p_{t\&i}} - \mathrm{mean}(\mathbf{p_{t\&i}}) \mid) > \alpha \end{cases} \quad (7)$$

where $\widehat{i}^t = \mathrm{mean}(\mathbf{p_t})$ and α is a threshold used to control the selection process (by increasing α we favor the temporal prediction). Adding an outlying value to the $\mathbf{p_t}$ vector will increase its mean absolute deviation, on the contrary an inlying value will maintain a similar mean absolute deviation. In our model we select temporal prediction when $\widehat{i}^i$ is an outlier, this corresponds to **Case 4**. For **Case 2** and **Case 3** we favor the temporal prediction and for **Case 1** we favor the inter-view prediction. The value for α used in this work was empirically found to be optimal at 0.5.

From this process, we deduce a binary selection map:

$$B(\mathbf{k}) = \begin{cases} 0 & \text{if } \widehat{i} = \widehat{i}^t \\ 1 & \text{if } \widehat{i} = \widehat{i}^i \end{cases} \quad (8)$$

which indicates the selected prediction type for each pixel.

E. Discussion on the method

In dense camera rig systems, a high number of views are available at the encoder side. Typically, only a subset is coded and sent in the bitstream, the rest being synthesized at the receiver side [20]. Our prediction method uses the synthesized view at the encoder side, since one frame per GOP of that view is transmitted in the bitstream. Indeed, synthesizing the intermediate views instead of sending them is a more efficient alternative as show in [47]. Our method can be seen as in between these two scenarios: we only send some information on the synthesized views, which we exploit to improve the synthesis. Consequently, in this work, we do not only propose a rendering method, but also a change in the design of the transmission stage. Note that we could have proposed a method where the key frames in the synthesized view are rendered with the left and right reference views using VSRS for instance, but then the rendering artifacts created in these key frames would be propagated to the rest of the frames in the motion compensation stage.

Furthermore, we use a "backward" motion compensation stage in our method: the vectors in $\mathbf{v}_s$ point from I^s_p (or I^s_f) to I^s_t. We can have a $\mathbf{v_s}$ that points from I^s_t to a past or future reference if the vectors in $\mathbf{v_r}$ point in the same direction (e.g., from I^r_t to I^r_p or I^s_f). This can easily be done if the inputs of the optical flow algorithm that outputs $\mathbf{v_r}$ are reversed. In this case, and if $\mathbf{k} = (x, y)$ is a point in I^r_t, Equation (2) becomes:

$$\mathbf{v}_s(\mathbf{k} + \mathbf{d}_t(\mathbf{k})) = \mathbf{v}_r(\mathbf{k}) + \mathbf{d}_{t-1}(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) - \mathbf{d}_t(\mathbf{k}) \quad (9)$$

From Equation (9), we can see that $\mathbf{v}_s$ is now defined for every pixel in I^r_t that has a correspondence in I^s_t. The holes in $\mathbf{v}_s$ (and in the corresponding prediction) correspond to disocclusions when warping from I^r_t to I^s_t. Even if we use a different time instant (f), the holes in the corresponding prediction would still come from warping I^r_t to I^s_t and thus will coincide with the holes of the first prediction. The merging process will not be able to fill in these holes and they will eventually have to be inpainted. In our method, the holes correspond to disocclusions when warping from I^r_p to I^s_p in the first prediction, and from I^r_f to I^s_f in the second. The holes do not necessarily coincide, and thus, pixels can be efficiently predicted from one or the other frame, during the merging process.

In comparison to other pixel-based methods such as [44] which improve the encoding of the depth map using dense MVFs computed on texture, our method warps the dense MVFs at the level of the intermediate view and uses them to motion compensate texture images as shown in this section.

Furthermore, boundary artifacts reduction methods such as [41] can be used in parallel with VSTP. Since, our final synthesis is a blend between DIBR rendering and the temporal predictions, reducing the artifacts in the DIBR synthesis will increase the quality of our method. Also, a better texture-depth alignment can benefit the warping of the dense MVFs. However, our method also gives the possibility to adjust the QP of the key frames which will in turn affect all frames inside a GOP or modify the frequency of the key frames which will reduce or increase the temporal distance of the prediction resulting in a higher quality rendering and a variation of the transmission rate.

As discussed above, our method provides new possibilities to control the rate and distortion in comparison to VSRS-1DFast: modifying the QP of key frames or adjusting their frequency. The bit allocation optimization scheme for DIBR multiview coding presented in [42] can be employed with our method as-well. However, a study towards the integration of the additional rate and distortion control options provided by VSTP within such schemes should be performed. For simplicity reasons in our experiments we will use the recommended depth and texture QPs for 3D-HEVC testing, as discussed in section IV-A.

IV. Experimental results

A. Experimental setting

Our algorithm takes as input two coded left and right views with their associated depth videos and camera parameters, and one frame per GOP of the intermediate view, and outputs the whole intermediate view after synthesizing the rest of the frames. The synthesis results are compared against the original intermediate sequences to measure the PSNR. We thus consider a five-view scenario in these experiments in which we code two views (left and right) and key frames from 1/2 view and synthesize three intermediary views at $1/4$, $1/2$ and $3/4$ positions between the two base views. We assume that one of the three intermediary views is available at the encoder side($1/2$). The coding configuration described in the Common Test Conditions (CTCs) defined by JCT-3V for conducting experiments with the reference software of 3D-HEVC [48] is used for coding the left and right views. The recommended texture and depth QPs are 25, 30, 35, 40 and 34, 39, 42, 45

respectively. The optical flow algorithm used in our method can be downloaded from [46], the configuration parameters are reported in Table I and more details can be found in [49].

TABLE I
OPTICAL FLOW PARAMETERS

Parameter	Description	Value
Alpha	Regularization weight	0.012
Ratio	Downsampling ratio	0.4
MinWith	Width of the coarsest level	20
nOuterFPIterations	Number of outer fixed point iterations	7
nInnerFPIterations	Number of inner fixed point iterations	1
nSORIterations	Number of Successive Over Relaxation iterations	30

We test our method on four sequences of the test set in the CTCs: Balloons, Kendo, Newspaper and PoznanHall2. Each sequence is composed of three real views and we also consider two virtual views. The CTCs indicate to use the middle view as base view, and the left and right views as dependent views. However, here we want the left and right views to be decodable without the middle view because only the first frame in each GOP of that view will be sent in the bitstream. We thus set the left view as base view, and the others as dependent views. Also, we code roughly 10 seconds of video of each sequences. Note that the number of frames is lower in PoznanHall2 because its frame rate is lower as well (cf. Table II).

TABLE II
SEQUENCES USED IN OUR EXPERIMENTS

Class	Sequence	Frames per second	Number of frames
class A (1920×1088)	PoznanHall2	25	200
class C (1024×768)	Balloons	30	300
	Kendo	30	300
	Newspaper	30	300

We compare our synthesis method to the reference VSRS-1DFast in 3D-HEVC test model, HTM. We evaluate the performance of the reference and the proposed methods using the Bjontegaard delta-PSNR (BD-PSNR) [50] metric on the synthesized views. The PSNR is evaluated against the original intermediate views. Evaluating our synthesis against frames synthesized from uncompressed views, as indicated by the CTCs, would penalize the lack of artifacts that arise from disparity warping, which are present in both compressed and uncompressed VSRS synthesis. The rate in the reference method is the sum of the rates needed to code the left and right views with their associated depth videos. The same rate is considered in the proposed method, to which is added the rate needed to code the first frame in each GOP of the intermediate view. We use the BD-PSNR metric to measure the improvement (see Figure 7).

B. Synthesis results

Table III gives the BD-PSNR values obtained with the two prediction schemes with simple merging ("Direct" and "Hierarchical") and "Adaptive Fusion" applied in the "Hierarchical" scheme ("HierarchicalAF") when considering only the PSNR of the 1/2 intermediary view synthesized with VSTP. In Table IV we show the BD-PSNR for the 3 intermediary views. Here, the PSNR is computed as the average between the 3 ($1/4$, $3/4$ synthesized with VSRS-1DFast and $1/2$ with VSTP). A positive value in this table indicates a gain. On average, our method brings 0.53dB, 0.59dB and 0.87dB BD-PSNR increase with "Direct" and "Hierarchical" schemes with simple temporal predictions merging, and the "Hierarchical" scheme with the "Adaptive Fusion" method respectively, compared to the reference VSRS-1DFast method. In the last column of the table (HierAF+HierSynth) we show the BD-PSNR obtained if we synthesize the $1/4$ and $3/4$ virtual views from left base view and our VSTP synthesis, and from VSTP synthesis and the right base view respectively. The depth map for the $1/2$ view is synthesized from right and left base views. By employing this hierarchical synthesis we take advantage of the higher quality of our rendering method to improve the $1/4$ and $3/4$ views without modifying the rate. The delta-PSNR between reference and ours for $1/4$ and $3/4$ views is -0.09dB, -0.01dB, 1.58dB for Balloons, Kendo and Newspaper sequences in average over all QPs. As expected these results are consistent with the BD-PSNR reported in Table IV(HierAF+HierSynth compared to HierarchicalAF), since the rate is not modified. Note, that the 5 view test case scenario no longer contains the Poznan Hall2 sequence. This is due to using original views as reference for evaluating the PSNR of the $1/4$ and $3/4$ views which in the case of Poznan Hall2 sequence are not available. As discussed in Section III-E synthesis is proven to be more efficient. However, the quality of an encoded view is always higher than that of a synthesis, we obtained 38.50dB PSNR compared to 35.81dB PSNR and 32.99dB PSNR for direct 3D-HEVC encoding, VSTP synthesis and VSRS-1DFast synthesis, respectively, in average over all sequences and all QPs.

TABLE III
BD-PSNR VALUES FOR A 3 VIEW TEST CASE, OBTAINED WITH BOTH PREDICTION SCHEMES AND ADAPTIVE FUSION IN THE PROPOSED METHOD COMPARED WITH THE REFERENCE VSRS-1D FAST METHOD.

Sequence	BD-PSNR (in dB)		
	Direct	Hierarchical	HierarchicalAF
Balloons	1.94	1.84	2.45
Kendo	-1.12	-0.56	0.93
Newspaper	4.70	4.80	5.28
PoznanHall2	2.17	1.99	2.32
Average	**1.92**	**2.01**	**2.74**

The Rate Distortion (RD) curves for the reference and the proposed method (for both schemes and merging methods) are given in Figure 7. We can see that while both schemes with simple merging outperform the reference method for Balloons and Newspaper, our method outperforms the reference only with the "Hierarchical" scheme with adaptive fusion in Kendo. This is also represented in BD-PSNR values for this sequence which are only positive in the "Hierarchical" scheme with adaptive fusion, as shown in Table III. Using the "Adaptive Fusion" method with the "Hierarchical" scheme brings high

TABLE IV
BD-PSNR VALUES FOR A 5 VIEW TEST CASE, OBTAINED WITH BOTH PREDICTION SCHEMES, ADAPTIVE FUSION AND HIERARCHICAL SYNTHESIS IN THE PROPOSED METHOD COMPARED WITH THE REFERENCE VSRS-1D FAST METHOD.

Sequence	**BD-PSNR (in dB)**			
	Direct	Hierarchical	HierarchicalAF	HierAF + HierSynth
Balloons	0.52	0.49	0.69	0.64
Kendo	-0.45	-0.27	0.22	0.22
Newspaper	1.52	1.55	1.71	2.78
Average	**0.53**	**0.59**	**0.87**	**1.21**

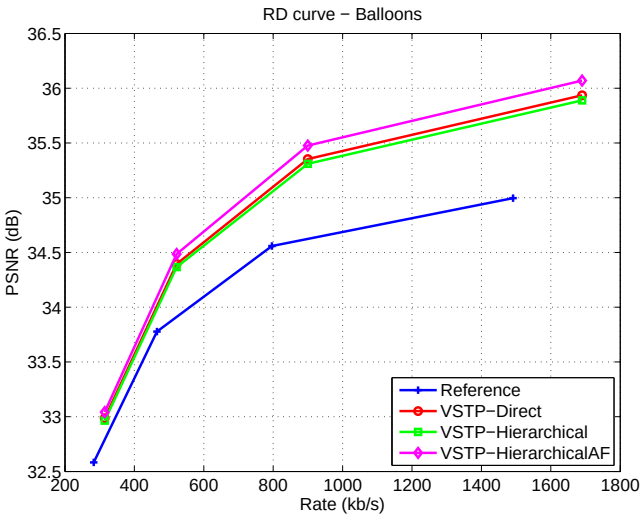
additional gains for Kendo sequence and moderate additional gains for Balloons, Newspaper sequences. This is expected because the fusion method was designed with the main goal of correcting bad temporal predictions caused by high intensity motion as is the case of Kendo sequence.

To better evaluate our method we perform an additional test. Since VSTP synthesis requires information to be sent through the bitstream, mainly one frame per GOP, we perform a direct comparison between the encoding of a dependant view and our VSTP synthesis. The results indicate we are able to outperform the encoding at low bitrates. This is possible due to encoding errors at lowbitrates having a greater impact on the quality of the image as compared to synthesis errors; while, at the same time synthesis provides better rate. The tests were performed on Balloons, Kendo and Newspaper sequences for QPs ranging from 50 to 35 and we obtained: 1.33, 1.061, 0.62 dB BD-PSNR gain, over 3D-HEVC, respectively for each sequence.

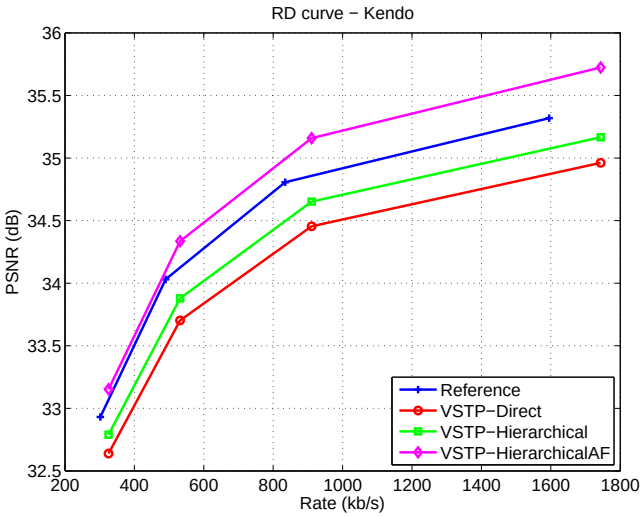
Figure 8 shows, for the four tested sequences, the variation of the PSNR of the synthesized view over time with the reference and the proposed method (both schemes and "Hierarchical" scheme with "Adaptive Fusion"). Only one QP (25) is represented for simplicity as the behavior of any curve is similar across all QPs. In the proposed method and for all sequences, we notice periodic peaks in the synthesized view PSNR, which correspond to the first frame of each GOP. Since these frames are not synthesized but rather decoded, their PSNR is higher than any other frames in the GOP. For the Balloons, NewspaperCC and PoznanHall2 sequences, the proposed method outperforms the reference VSRS-1DFast rendering for most frames. For the Kendo sequence, our method is better only in certain parts.

Figure 9 shows two side-by-side examples of ideal and real fusion maps for Kendo and NewspaperCC sequences. The ideal fusion map displayed here is only showing, in black, the pixels that if replaced by inter-view prediction, would have their absolute error decreased by at least 5 (we ignore small gains). We can see that our map is consistent with the ideal map for correcting high errors. This is also shown in Figure 10 where we display the difference between the absolute error of temporal and inter-view prediction for the same frame of Kendo sequence. Positive values indicate inter-view prediction is better and we can see a correspondence between high values and our fusion map.

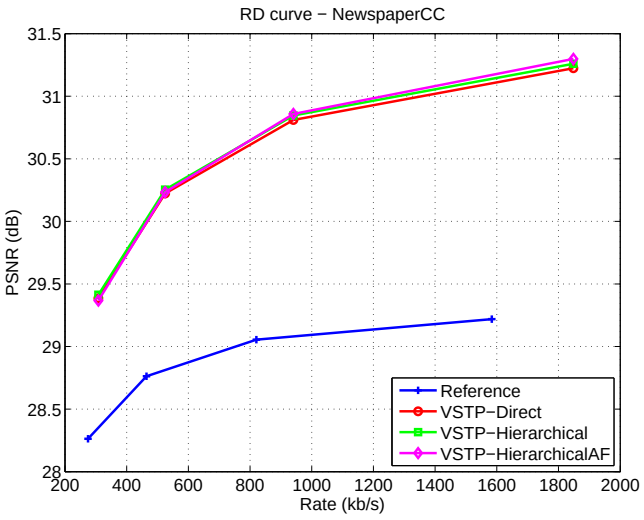
Figure 11 shows parts of frames synthesized using the ref-



(a) Balloons

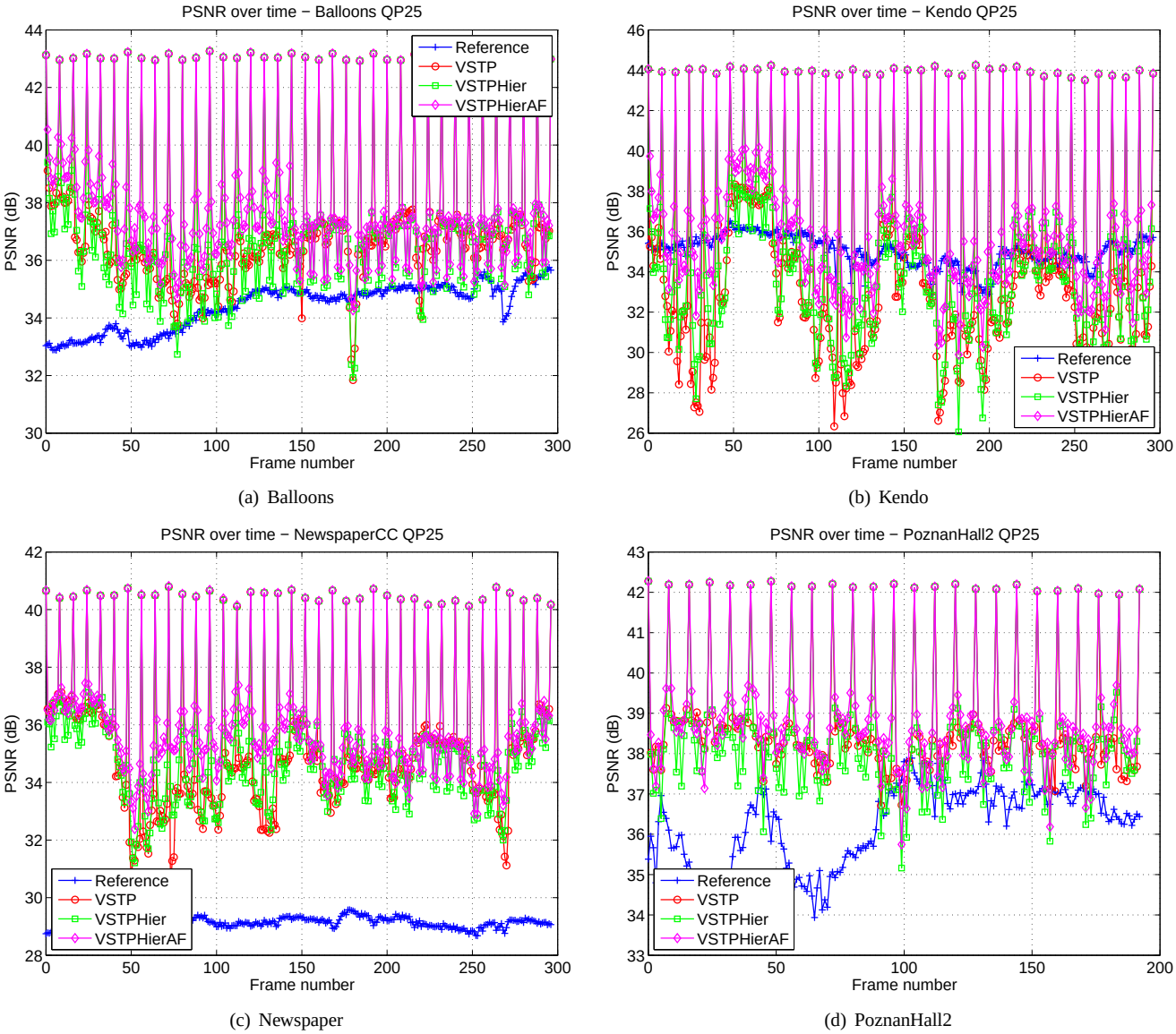


(b) Kendo



(c) Newspaper

Fig. 7. RD curves of the reference and proposed method on 5 view test scenario for the Balloons, Kendo and NewspaperCC sequences.



(a) Balloons (b) Kendo

(c) Newspaper (d) PoznanHall2

Fig. 8. Variation of the PSNR of the middle synthesized view over time for the reference and proposed method at QP 25.

erence and the proposed method with hierarchical scheme and Figure 12 shows parts of frames using the proposed method with and without adaptive fusion. For fairness of comparison, for our method, we show frames that are actually synthesized and not decoded. We can notice a clear improvement in the synthesis quality with our method: the artifacts obtained with VSRS-1DFast (highlighted in red in the figures) are efficiently removed and also artifacts in our method are removed when using the adaptive fusion.

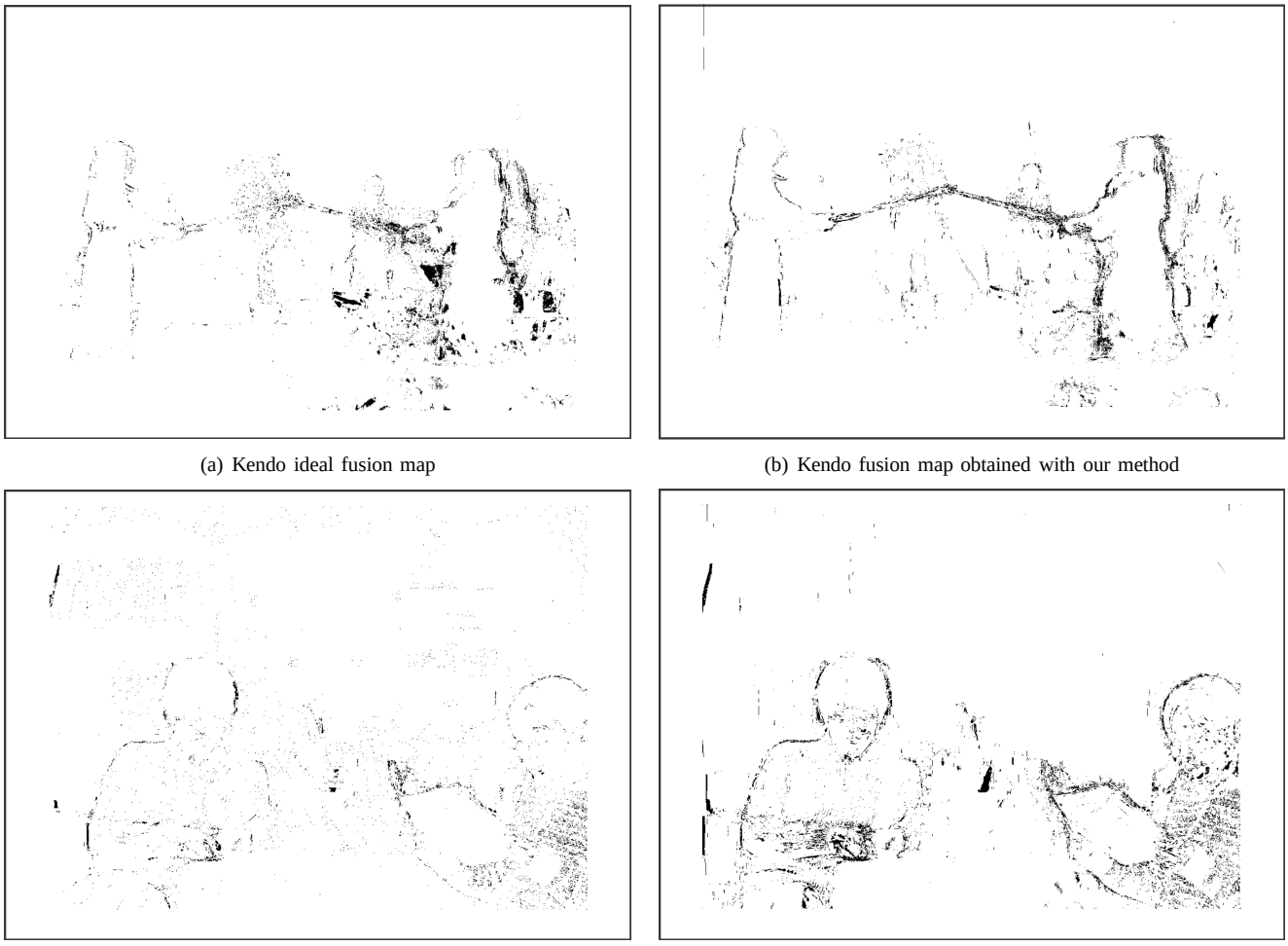
C. Results interpretation

The "Adaptive Fusion" method with the "Hierarchical" scheme brings high gains in BD-PSNR. To better describe our results we will refer to an ideal case where we use the original frames to create a fusion map in which we mark all the pixels that have a lower error in the inter-view prediction compared to the temporal one, for simplicity we will only test 3 seconds from each sequence. As a mean of verifying the quality of our obtained fusion map we compute the difference between the mean absolute error (MAE) of pixels marked by a fusion map, for temporal and inter-view predictions, referred to as ΔMAE as shown in the following equation, where $\widehat{I}$ is either the temporal or inter-view prediction, B is the binary fusion map and $\widehat{I}_t$, $\widehat{I}_i$ and I are the temporal and inter-view predictions and the original frame respectively.

$$\mathrm{MAE}(\widehat{I}, B) = \begin{cases} 0, & \text{if } B(x,y) = 0 \quad \forall \quad x, y \\ \dfrac{\sum_{x=1}^{M}\sum_{y=1}^{N} B(x,y)|\widehat{I}(x,y) - I(x,y)|}{\sum_{x=1}^{M}\sum_{y=1}^{N} B(x,y)}, & \text{otherwise} \end{cases}$$

$$\Delta\mathrm{MAE}(\widehat{I}_t, \widehat{I}_i, B) = \mathrm{MAE}(\widehat{I}_t, B) - \mathrm{MAE}(\widehat{I}_i, B) \tag{10}$$

Table V shows the percentages of replaced pixels and the MAE reduction for our method and the ideal case. The values in Table V are the averages for all QPs. For example let us



(a) Kendo ideal fusion map

(b) Kendo fusion map obtained with our method

(c) Newspaper ideal fusion map

(d) Newspaper fusion map obtained with our method

Fig. 9. Fusion maps for frame 4 in Kendo and Newspaper sequences, at QPs 30 and 25 respectively. Pixels in the temporal prediction that are replaced with inter-view prediction are black. Figures 9(a) and 9(c) are the ideal maps in which inter-view prediction is only selected if it corrects high temporal errors (the original view was used for this computation). Figures 9(b) and 9(d) are obtained with the "Adaptive Fusion" method.

TABLE V
ADAPTIVE FUSION RESULTS: PERCENTAGE OF REPLACED PIXELS AND MAE GAINS FOR OUR METHOD AND THE IDEAL CASE IN WHICH THE FUSION MAP IS DETERMINED USING THE ORIGINAL VIEW.

Sequence	**Inter-view predicted pixels (%)**		**ΔMAE**	
	Real	Ideal	Real	Ideal
Balloons	2.74	30.58	0.50	2.57
Kendo	3.67	27.13	2.20	3.48
Newspaper	3.16	28.39	0.58	7.35
PoznanHall2	0.81	26.05	-0.55	2.34
Average	**2.60**	**28.03**	**0.68**	**2.65**

consider the Kendo sequence at QP 25. In average for this case 25.39% of the pixels in a frame are better predicted with inter-view prediction, our method selects 3.48% of the pixels to be replaced by inter-view prediction, out of which 1.6% is a bad selection (temporal prediction was actually giving better results and we replaced it with inter-view prediction). Note that the 25.39% ideally selected pixels include predicted areas which are better only by a small margin. Our selection however focuses on correcting high errors. Even though parts of our replaced areas are actually worse predictions and increase the MAE, overall we still obtain a positive ΔMAE which shows we are correcting the high errors, as also shown in Figures 9 and 10. For the Balloons and Newspaper sequences where the introduction of "Adaptive Fusion" brings a small additional increase in BD-PSNR we have a smaller percentage of replaced pixels with a small ΔMAE in contrast to the Kendo sequence where this method brings a high additional increase in BD-PSNR. For the PoznanHall2 sequence we have a similar result in BD-PSNR, the "Direct" and "Hierarchical" schemes already provide a very good result due to low intensity motion. Here the "Adaptive Fusion" method corrects some small temporal prediction errors but also introduces inter-view prediction errors, this explains why we have a negative ΔMAE over the replaced pixels in this sequence. Note that the number of replaced pixels is smaller compared to the other sequences, only 0.81% of a frame on average, thus the quality of the entire image is affected only by a small margin.

The results of Table III and the RD curves in Figure 7 show that the "Hierarchical" scheme outperforms the "Direct"

scheme, which was expected, since the temporal prediction distances are shorter in the first scheme. Note that in a GOP of 8 frames, the fifth frame is synthesized in the same way in both shemes, which is why the curves of Figure 8 corresponding to the two schemes, intersect not only in the first frame of each GOP but also in the fifth frame.

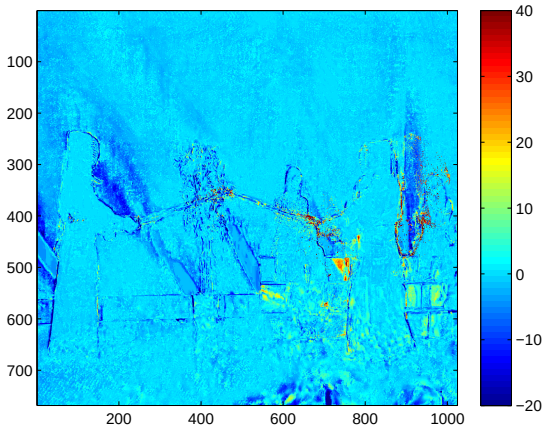
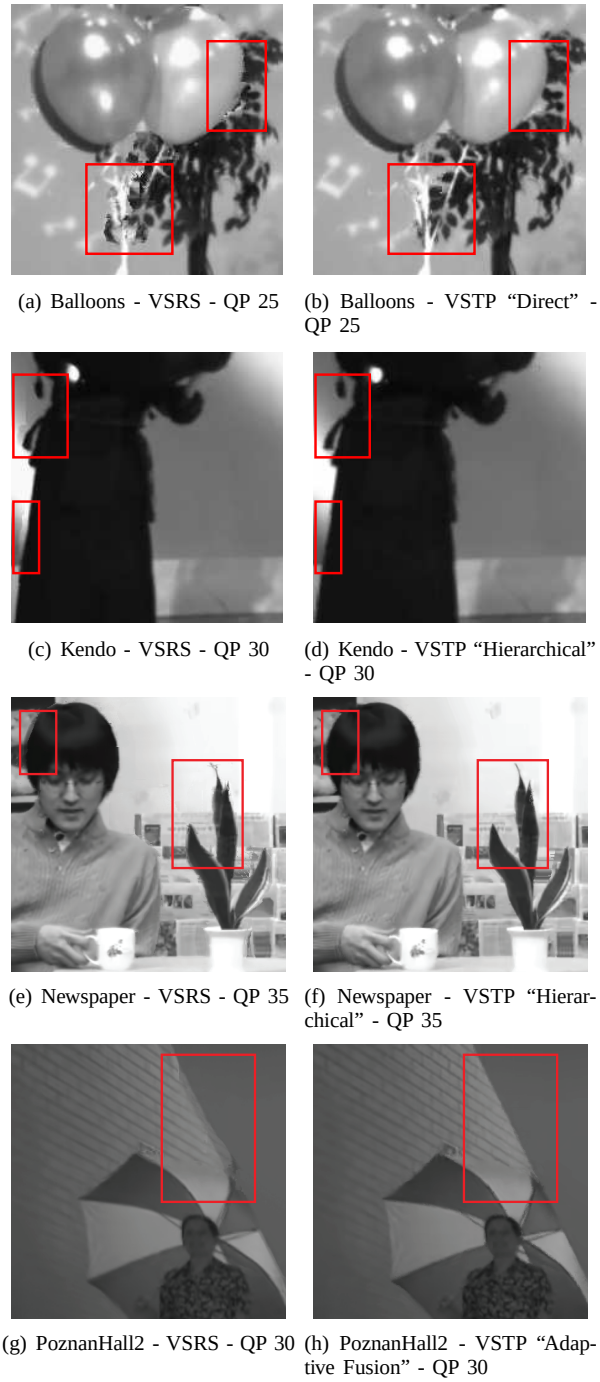


Fig. 10. Difference between inter-view and temporal prediction error (ΔMAE) on frame 4 in Kendo sequence, QP 30.

Our method improves the quality of the synthesis on three levels: first, it accounts for a difference in illumination between the coded reference views and the synthesized view, which rendering techniques such as VSRS-1DFast cannot do. Indeed, while VSRS-1DFast cannot warp a different illumination level from the reference views into the synthesized view, our method propagates the correct illumination level of the sent key frames accross the rest of the frames using motion compensation. Second, our method fills holes due to disocclusions more efficiently than VSRS-1DFast. Indeed, these holes are filled using inpainting in the latter, hence creating artifacts such as the ones highlighted in Figure 11. In our method, the disocclusion areas can be found in previously synthesized frames. Third, foreground objects are better rendered because the method is less sensitive to depth distortions. We use disparity to warp dense MVFs rather than directly warping the texture (cf. Figures 11(e), 11(f), 11(g), 11(h)). In addition, VSTP brings texture information from different time instants that cannot be obtained from inter-view prediction. The fusion between the two prediction types will reduce the chance of having residual holes in the final synthesis. This explains how our method efficiently removes the aforementioned artifacts, as shown in Figure 11. Also, subjective viewing of the sequences has shown that there are no flickering effects with our method. A synthesis example can be downloaded for viewing at the following links:
http://perso.telecom-paristech.fr/~cagnazzo/vsrs.zip
http://perso.telecom-paristech.fr/~cagnazzo/vstp.zip
for VSRS-1DFast and VSTP respectively.

Our method is inherently more complex than VSRS-1DFast due to the dense motion estimation / compensation stage. Shortcuts that can reduce the complexity of our method, at the



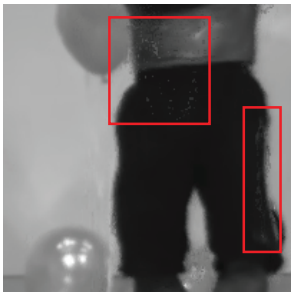
(a) Balloons - VSRS - QP 25 (b) Balloons - VSTP “Direct” - QP 25

(c) Kendo - VSRS - QP 30 (d) Kendo - VSTP “Hierarchical” - QP 30

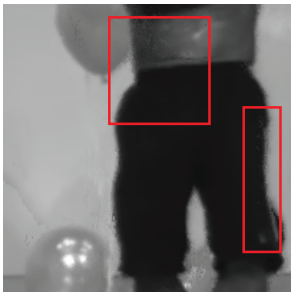
(e) Newspaper - VSRS - QP 35 (f) Newspaper - VSTP “Hierarchical” - QP 35

(g) PoznanHall2 - VSRS - QP 30 (h) PoznanHall2 - VSTP “Adaptive Fusion” - QP 30

Fig. 11. Parts of frames synthesized with the reference VSRS-1DFast and the proposed method. Highlighted artifacts in VSRS-1DFast (Figures 11(a), 11(c), 11(e) and 11(g)) are efficiently removed in our method (Figures 11(b), 11(d), 11(f) and 11(h)).



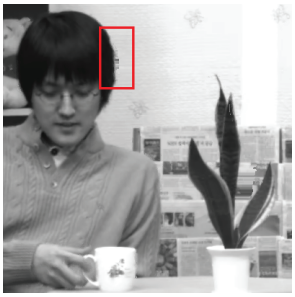
(a) Balloons - VSTP - QP 25



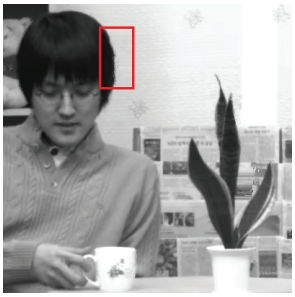
(b) Balloons - VSTP "Adaptive Fusion" - QP 25

(c) Kendo - VSTP - QP 30

(d) Kendo - VSTP "Adaptive Fusion" - QP 30



(e) Newspaper - VSTP - QP 25



(f) Newspaper - VSTP "Adaptive Fusion" - QP 25

(g) PoznanHall2 - VSTP - QP 30

(h) PoznanHall2 - VSTP "Adaptive Fusion" - QP 30

Fig. 12. Parts of frames synthesized with and without "Adaptive Fusion". Highlighted artifacts after merging the temporal predictions (Figures 12(a), 12(c), 12(e) and 12(g)) are efficiently removed when using "Adaptive Fusion" (Figures 12(b), 12(d), 12(f) and 12(h)).

price of loosing some prediction accuracy, include block-based motion estimation/compensation and uni-predictive motion compensation (predict using only a past frame, or only a future frame).

V. Conclusion and Future Work

In this paper, we presented a view synthesis technique that exploits temporal prediction in order to improve the quality of the synthesis. Namely, some key frames of the synthesized view are encoded in the bitstream, and the rest are interpolated using motion compensation with vectors warped from reference views. Four predictions using the left and right reference view, and a past and future time instant can be constructed and then merged together into a single prediction of the synthesized frame. Two prediction schemes referred to as "Direct" and "Hierarchical" have been presented in this work. The first synthesizes frames using motion compensation only with key frames, while the other motion compensates with previously synthesized frames, hence reducing the prediction distances. We also introduced a prediction merging method referred to as "Adaptive Fusion" that selects between inter-view and temporal prediction, thus removing some of the motion estimation errors. Our method brings 0.53dB and 0.59dB PSNR increase with the "Direct" and "Hierarchical" schemes respectively and 0.87dB PSNR with "Hierarchical" scheme and "Adaptive Fusion" in average for several test sequences over the state-of-the-art VSRS-1DFast software under 3D-HEVC standards. Furthermore, the MVF precision on frames with high intensity motion can be improved by using a better motion estimation technique or using an adaptive GOP size with respect to motion intensity. The "Adaptive Fusion" method can be further improved by finding a better inter-view/temporal selection criterion. Additional adjacent views that are not available at the encoder side can be further improved by deriving the vector fields required to directly predict the frames from the key frames. Finally, the frequency at which key frames are sent in our method, which, in the current version, follows the GOP structure used for coding the reference views, can be modified: lower frequencies allow bitrate savings since less key frames will be sent but they also imply motion estimation between distant frames, which will decrease the prediction accuracy. Finding a good trade-off for this parameter is an interesting future research subject.

References

[1] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, Eds., *Emerging technologies for 3D video: content creation, coding, transmission and rendering*. Wiley, May 2013.

[2] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-Viewpoint TV," *IEEE Signal Processing Magazine*, vol. 28, pp. 67–76, 2011.

[3] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," *IEEE International Conference on Image Processing*, vol. 1, pp. 201–204, 2007.

[4] C. Fehn, "A 3D-TV approach using depth-image-based rendering," in *3rd IASTED Conference on Visualization, Imaging, and Image Processing*, Benalmadena, Spain, 8-10 September 2003, pp. 482–487.

[5] H. Shum and S. B. Kang, "Review of image-based rendering techniques," *SPIE Visual Communications and Image Processing*, vol. 4067, pp. 2–13, 2000. [Online]. Available: http://dx.doi.org/10.1117/12.386541

[6] L. Zhan-Wei, A. Ping, L. Su-xing, and Z. Zhao-yang, "Arbitrary view generation based on DIBR," in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Xiamen, People's Republic of China, 2007, pp. 168–171.

[7] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *19th International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[8] F. Dufaux, M. Cagnazzo, and B. Pesquet-Popescu, *Motion Estimation - a Video Coding Viewpoint*, ser. Academic Press Library in Signal Processing, R. Chellappa and S. Theodoridis, Eds. Academic Press, 2014 (to be published), vol. 5: Image and Video Compression and Multimedia.

[9] R. Krishnamurthy, P. Moulin, and J. Woods, "Optical flow techniques applied to video coding," in *IEEE International Conference on Image Processing (ICIP)*, vol. 1, 1995, pp. 570–573 vol.1.

[10] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of SIGGRAPH*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 31–42. [Online]. Available: http://doi.acm.org/10.1145/237170.237199

[11] C. Buehler, M. Bosse, L. McMillan, and S. Gortler, "Unstructured Lumigraph Rendering," in *Proc SIGGRAPH*, Los Angeles, California USA, August 2001, pp. 425–432.

[12] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics," in *Proceedings SIGGRAPH*, Los Angeles, California USA, 1999, pp. 299–306.

[13] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *IEEE MMSP*, Saint Malo, France, 4-6, October 2010.

[14] C. Guillemot and O. L. Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, pp. 127–144, 2014.

[15] M. Bertalmio and G. Sapiro, "Image inpainting," in *SIGGRAPH*, New Orleans, USA, Jully 2000, pp. 417–424.

[16] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[17] "High Efficiency Video Coding," ITU-T Recommendation H.265 and ISO/IEC 23008-2 HEVC, April 2013.

[18] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP Journal on Advances in Signal Processing*, 2009.

[19] "Report on experimental framework for 3D video coding," ISO/IEC JTC1/SC29/WG11 MPEG2010/N11631, October 2010.

[20] L. Zhang, G. Tech, K. Wegner, and S. Yea, "3D-HEVC test model 5," ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JCT3V-E1005, July 2013.

[21] "w14551 - Description of Exploration Experiments on Free-Viewpoint-Television (FTV)," MPEG, Sapporo meeting, july, 2014,.

[22] S. Shimizu and H. Kimata, "Improved view synthesis prediction using decoder-side motion derivation for multiview video coding," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2010, pp. 1–4.

[23] K.-Y. Chen, P.-K. Tsung, P.-C. Lin, H.-J. Yang, and L.-G. Chen, "Hybrid motion/depth-oriented inpainting for virtual view synthesis in multiview applications," *3DTV-CON*, pp. 1–4, 7-9 June 2010.

[24] W. Sun, O. C. Au, L. Xu, Y. Li, and W. Hu, "Novel temporal domain hole filling based on background modeling for view synthesis," in *IEEE International on Image Processing (ICIP)*, Orlando, FL, 30 Sept. - 3 Oct. 2012, pp. 2721 – 2724.

[25] K. P. Kumar, S. Gupta, and K. S. Venkatesh, "Spatio-temporal multi-view synthesis for free viewpoint television," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, Aberdeen, 7-8 October 2013, pp. 1 – 4.

[26] K.-W. H. W.-C. Siu, "Depth-assisted nonlocal means hole filling for novel view synthesis," in *ICIP*, September 2012.

[27] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression," *PCS 2006*.

[28] Y. Morvan, D. Farin, and P. H. N. de With, "Multiview depth-image compression using an extended h.264 encoder," *LNCS*, 2007.

[29] S. Yea and A. Vetro, "View synthesis prediction for multiview video codings," *Elsevier, Signal Processing: Image Communication*, January 2009.

[30] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding of free viewpoint video for multi-path network streaming," in *IEEE Globecom*, December 2012.

[31] H. Yuan, J. Liu, Z. Li, and W. Liu, "Virtual view synthesis for 3d video system: Theoretical analyses and implementation," *IEEE Transactions on Broadcasting*, vol. 58, pp. 558–568, Mars 2012.

[32] J. Kim, Y.-G. Kim, H. Song, T.-Y. Kuo, and Y. J. Chung, "TCP-friendly internet video streaming employing variable frame-rate encoding and interpolation," *CSVT*, October 2000.

[33] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *CSVT*, April 2007.

[34] C. Wang, L. Zhang, Y. He, and Y.-P. Tan, "Frame rate up-conversion using trilateral filtering," *CSVT*, June 2010.

[35] E. G. Mora, "Multiview video plus depth coding for new multimedia services," Ph.D. dissertation, chapter 6, pg. 120, EDITE, Telecom Paristech, 2014.

[36] G. Tech, "HTM-7.0 software," Available: https://hevc.hhi.fraunhofer.de/.

[37] M. S. Farid, M. Lucenteforte, and M. Grangetto, "Depth image based rendering with inverse mapping," in *IEEE MMSP*, Pula (Sardinia), Italy, September 30-October 2, 2013, pp. 135–140.

[38] P.-J. Lee and Effendi, "Adaptive edge-oriented depth image smoothing approach for depth image based rendering," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Shanghai, 24-26 March, 2010, pp. 1–5.

[39] L. Wang, J. Liu, J. Sun, Y. Ren, W. Liu, and Y. Gao, "Virtual view synthesis without preprocessing depth image for depth image based rendering," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Antalya, Turkey, 16-18 May 2011, pp. 1–4.

[40] Z. Wang and J. Zhou, "A novel approach for depth image based rendering, based on non-linear transformation of depth values," in *International Conference on Image Analysis and Signal Processing (IASP)*, Hubei, People's Republic of China, 21-23 October 2011, pp. 138–142.

[41] Y. Zhao, C. Zhu, Z. Chen, D. Dian, and L. Yu, "Boundary artifact reduction in view synthesis of 3d video: from perspective of texture-depth alignment," *IEEE Transactions on Broadcasting*, vol. 57, pp. 510–522, 2011.

[42] G. Cheung, V. Velisavljevic, and A. Ortega, "On dependent bit allocation for multiview image coding with depth-image-based rendering," *IEEE Transactions on Image Processing*, vol. 20, November 2011.

[43] J. Xiao, M. Hannuksela, T. Tillo, M. Gabbouj, C. Zhu, and Y. Zhao, "Scalable bit allocation between texture and depth views for 3d video streaming over heterogeneous networks," *IEEE Trans. Circuits and Systems for Video Technology*, p. 1, Juin 2014.

[44] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-based inter prediction in coded texture assisted depth coding." *IEEE Signal Processing Letters*, vol. 21, pp. 74–78, 2014.

[45] I. Daribo, W. Milded, and B. Pesquet-Popescu, "Joint Depth-Motion Dense Estimation for Multiview Video Coding," *Journal of Visual Communication and Image Representation*, vol. 21, pp. 487–497, 2010.

[46] C. Liu. Optical flow Matlab/C++ code. [Online]. Available: http://people.csail.mit.edu/celiu/OpticalFlow/

[47] E. G. Mora, "Multiview video plus depth coding for new multimedia services," Ph.D. dissertation, chapter 1, pg. 4, EDITE, Telecom Paristech, 2014.

[48] D. Rusanovsky, K. Muller, and A. Vetro, "Common Test Conditions of 3DV Core Experiments," ITU-T SG16 WP3 & ISO/IEC JTC1/SC29/WG11 JCT3V-D1100, April 2013.

[49] C. Liu, "Beyond pixels: exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, May 2009.

[50] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, April 2001.