CONSISTENT VIEW SYNTHESIS IN INTERACTIVE MULTIVIEW IMAGING

Thomas Maugey, Pascal Frossard

Signal Processing Laboratory (LTS4)
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Gene Cheung

National Institute of Informatics
Tokyo, Japan

ABSTRACT

An important question in the design of interactive multiview systems consists in determining the information needed by the decoder for high quality navigation between the views. Most of the existing techniques focus on the captured sequences and only consider their transmission, which does not guarantee consistency among receiver-generated frames of chosen virtual views. In this work, we propose a solution that additional transmits auxiliary information in order to help the construction of synthesized views, especially in the occluded areas. Comparative results with existing approaches validate this novel representation of multiview data for interactive navigation. We show that decoding quality and consistency among frames are improved with only a small share of additional information.

Index Terms— 3D video coding, interactive navigation, virtual view synthesis

1. INTRODUCTION

Interactive multiview video streaming is a novel paradigm in multiview video processing that is very challenging since it offers the possibility to change viewpoints in real-time. The users may thus navigate in different views corresponding to the original camera pictures, or additional virtual views that are created at the decoder for increasing the navigation capacities and the look-around effect [1], *i.e.* the sensation of immersion in the scene. The virtual view synthesis process is usually performed in two steps: projection of pixels between different viewpoints using depth information [2] (depth-image-based rendering, DIBR), and filling of the holes due to occlusions with inpainting techniques [3]. Therefore, interactive multiview video streaming requires to develop specific coding strategies that differ from the classical multiview video coding approaches [4].

Video encoding solutions for interactive systems mainly rely on the idea of adapting the prediction structure between the frames: real-time encoding [5], GoGOP design [6], considering different types of prediction and frame description [7]. Although these works offer interesting performance in the representation of the camera views, the virtual view generation problem is not really solved nowadays. This mainly comes from the occlusion filling techniques that are lacking information for properly reconstructing incomplete areas after DIBR. Even if synthesis techniques are able to build a good visual quality estimation of the occluded regions based on the available part of the synthesized image, they generally omit to consider the adjacent frames in the reconstruction. This leads to different reconstruction results in consecutive images and generates flickering effects in the user viewing experience. Since the flickering effect is percieved as the most annoying noise in video quality [8], some works have proposed to handle such artifact by performing the hole reconstruction jointly with the adjacent frames [9, 10]. These solutions are effective in a classical multi-view transmission framework where the decoder disposes of the adjacent images, but they are not appropriate in an interactive system where only a subset of requested frames are transmitted to the client. This is especially the case if, as in our work, the server transmits one reference view (color and depth) for the generation of the virtual images, instead of two as in most of the systems.

In this work we propose a novel paradigm where the classical information containing color and depth data is complemented with additional auxiliary information (AI) for effective view synthesis. This AI is constructed for a good representation of the information that is typically missing in the reconstruction of synthetic views, such as information about occluded areas. In other words, thanks to this AI, the inpainting techniques at the receiver can estimate the occluded regions of the virtual frames with consistency between adjacent frames. Experimental results show that this AI allows client's view synthesis process to significantly decrease the flickering noise with respect to the classical synthesis methods. We also show that the additional cost of this AI is very limited compared to a simple (but widely adopted) solution that consists in sending two reference views to enable view synthesis at the decoder. This observation opens new possibilities in interactive multiview streaming and video coding.

2. AUXILIARY INFORMATION TRANSMISSION

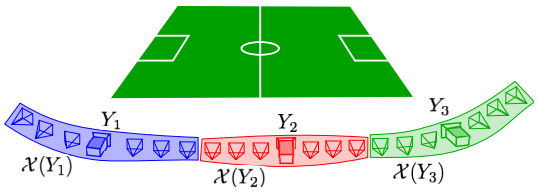


Fig. 1. Neighborhood definition

We assume that the user intends to observe a scene from different viewpoints. This *scene*, S, can be modeled as a countable set of random variables, s_i, taking their values in $\mathcal{V}$. These random variables correspond to the different points of the scene and the random values give the color information (typically $\mathcal{V} = \mathbb{R}^3$). The scene S is captured by different cameras producing at every instant a set of images. We define an *image*, Y, as is a finite set of N random variables, $\{y_i\}_{i=1,\dots,N}$ that take their values in $\mathcal{V}$. The link between the scene and a captured image Y is given with the *projection function*,

f_Y:

$$f_Y : Y \to S$$
$$y \to s = f_Y(y)$$

These projection functions are related to the geometry of the scene. In practice, they are defined by the depth of the scene and the parameters of the cameras. An image Y does not capture every elements of the scene, this is why we define $S_Y = f_Y(Y)$, the finite set of elements of S mapped to Y. This corresponds to the set of elements of S that are visible in image Y. If Y' is another captured image we say that Y and Y' are *geometrically correlated* if $S_Y \cap S_{Y'} \neq \emptyset$.

Let us assume now that the scene is captured by N_{ref} cameras obtaining at every instant a set of *reference frames*, $\{Y_i\}_{i=1\dots N_{ref}}$. Between each reference frame the system enables the users to navigate in virtual viewpoints which corresponds to *virtual* images X_j. We assume that every virtual viewpoint is attached to one and only one reference camera (simply the closest one). So we define the set of virtual images attached to one reference image Y as the *neighborhood of Y*, $\mathcal{X}(Y)$ (an example of neighborhood definition is given in Fig. 1). The generation of a virtual image is then performed in two steps. Let X be a virtual image of $\mathcal{X}(Y)$. To estimate X we first project the elements of Y and we obtain part of the image X: $f_X^{-1}(S_Y) = \{x \in X | f_X(x) \in S_Y\}$ called $X|Y$. Generally $X \setminus (X|Y) \neq \emptyset$ because of occlusions in the scene. This set, $X \setminus (X|Y)$, is called the *innovation* of X with respect to Y and needs to be estimated using inpainting algorithms [11]. The purpose of these inpainting algorithms is to recover the real points of the scene $S_X \setminus S_Y$. Since the innovations of different frames of the neighborhood can have pixels in common, we consider the *neighborhood innovation*

$$\Phi = \bigcup_{X \in \mathcal{X}_\Delta(Y)} S_X \setminus S_Y,$$

which corresponds to the total innovation brought by the virtual frames X of a neighborhood $\mathcal{X}_\Delta(Y)$ (schematized in Fig. 2). In our system, the proposed AI actually describes this Φ set. In other words, Φ is estimated at the encoder (that has the knowledge of every frames) and is transmitted in order to help synthesis at decoder which only possesses Y. Finally, Φ may be represented globally for different time instants in order to describe the innovations over a whole period of time, which reduces the cost of AI transmission.

3. SYSTEM DESCRIPTION

In the proposed system, the user interacts with a server by indicating its position every N_T frames. Based on this knowledge, the server sends the appropriate data to the decoder. More precisely, the user receives data corresponding to the neighborhoods it is navigating in so that he could freely change his viewpoint by creating synthetic views until he sends the next message to the server. For every considered neighborhood, the server sends the reference images Y that are coded with classical mono-view compression techniques [12]. In addition, the servers transmits the depth information that is used at the receiver to build the projection functions f_Y and f_X ($X \in \mathcal{X}(Y)$). In practice, the server sends depth information related to the Y image and the decoder uses it to estimate the general geometry of the other viewpoints which enables view synthesis by projection of pixels from the reference view. Finally, the server sends the AI that enables the user navigation in all the frames X of the neighborhood. More precisely, the AI is coded as a hash information: instead of transmitting the whole Φ set, we only send $\boldsymbol{\varphi} = h(\boldsymbol{\Phi})$, where $\boldsymbol{\Phi}$ is vector associated to Φ.

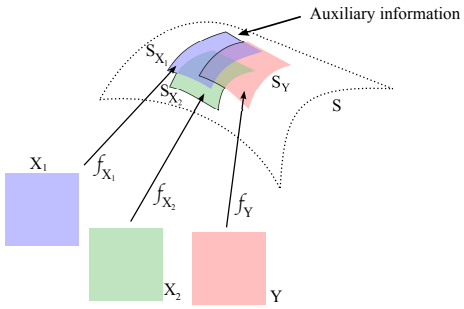


Fig. 2. Auxiliary information.

Since $\boldsymbol{\varphi}$ has been built as a hash information, the decoder constructs $\hat{\Phi}$ such as $h(\hat{\boldsymbol{\Phi}}) = \boldsymbol{\varphi}$. In this work we propose a reconstruction strategy that is based on the Criminisi's inpainting algorithm [11]. The latter technique is made of two steps. During the first one the algorithm chooses the patch that has the higher priority based on image gradient considerations. The second step fills the selected patch by using another similar patch from the image. We modify the inpainting algorithm by introducing a hash validation at this last step. The hole filling technique thus chooses a patch that corresponds to the AI. It is important to note that the design of the AI coding technique does not depend on the decoder. In parallel, the reconstruction techniques is independent from the type of hash information sent.

In this work, we propose three techniques to build this $\boldsymbol{\varphi}$ vector. The first construction of $\boldsymbol{\varphi}$ consists in transmitting a down-sampled version of $\boldsymbol{\Phi}$. This method is called DS in the following. The second approach is related to the classical hash information as proposed in the channel coding schemes [13]. For each sub-vectors of $\boldsymbol{\Phi}$, we transmit check sum bits that are calculated by adding the elements of the sub-vector and the sum is represented over a predefined number of bits. This approach is called CS in the following. Finally we test a solution that consists in transmitting a quantized version of low-band frequency coefficients of a DCT transform of sub-vectors of $\boldsymbol{\Phi}$. This is denoted by the DC solution in the following. For these three methods, the compression of the $\boldsymbol{\varphi}$ vector is done using a classical arithmetic coder.

4. EXPERIMENTS

We present here some experiments that illustrate the interest of transmitting additional information. We show that our novel representation significantly improves the consistency of the virtual frames with a reasonable additional cost. We also show that our solution performs better than a baseline technique that would consist in transmitting two reference views. We have built two illustrative example sequences, *rect 1* and *rect 2*. They are both made of two views that are capturing a background with a moving gray rectangle at the foreground (the type of background differs from one sequence to another), the first image of each views are shown in Fig 3. We later provide some results with the well-known natural sequence *ballet*. In the following, we consider that the set of achievable views is composed by N equidistant virtual viewpoints linearly arranged between the two cameras. The same AI is transmitted in order to permit the navigation on all these views. We note that, by construction, the AI's size depends on the distance between the reference cameras and not directly on the N value, which means that we can increase the

(a) *rect 1* sequence (b) *rect 2* sequence

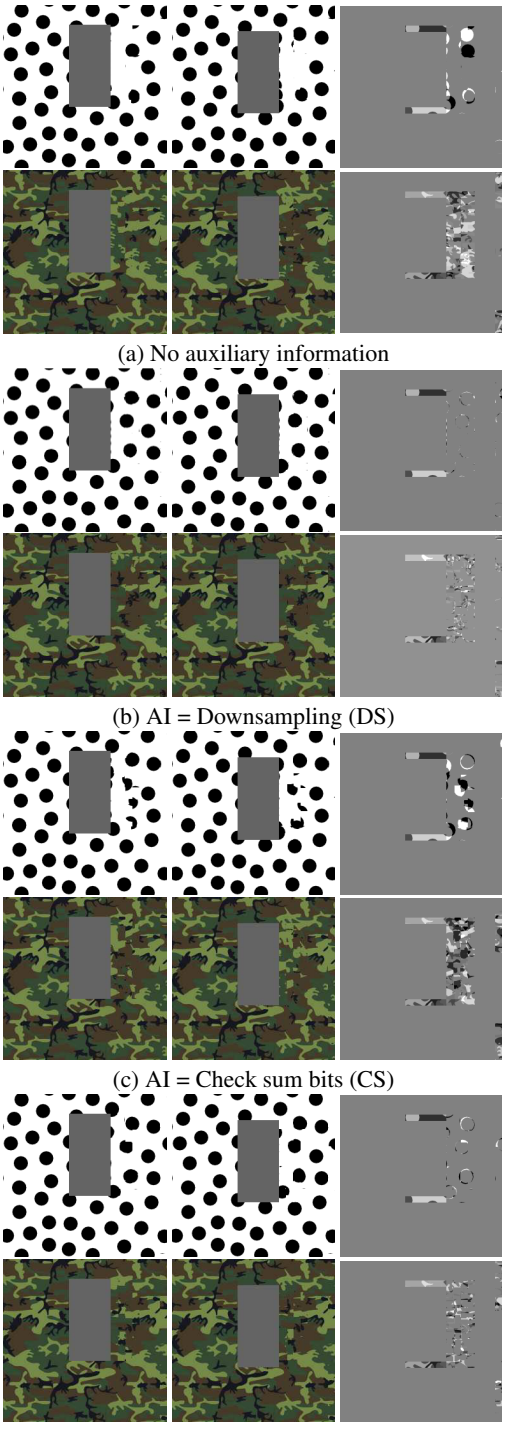
Fig. 3. First frame of the toy-example sequences. In the next frames the rectangle is moving from up to down.

number of intermediate views without decreasing the coding performance.

In order to validate the quality of the reconstruction, we show, in Fig. 4, some visual results on two synthesized images at two consecutive instants for different methods of AI construction. We first project the reference images of view 1, that are compressed at high quality level (similar results have been obtained at different compression levels). Then we perform the hole filling algorithm using the AI information (Φ vectors are compressed with a lossless arithmetic coder). The first two images of each line correspond to these two estimations. The third image presents the difference between these two frames. We observe in Fig. 4 that sending no additional information leads to two completely different consecutive frames. Indeed, we can see high errors (white and black) in the difference images. It seems that the CS approach obtains a similar low quality (Fig. 4 (c)). However, the DS and DC approaches, in Fig. 4 (b) and (d), manage to reconstruct the background with a good consistency between the two consecutive images.

In order to measure the cost of the AI in terms of bit rate, we calculate the rate of encoding the reference sequence and the AI for the different methods considered above. We compress the reference view (color and depth) with H.264 at different rate-distortion (RD) points (QP for H.264). Then, we add the obtained rate with the AI's rate which does not vary with the level of reference view compression, since the AI was estimated at the encoder with the original images and was compressed with a lossless arithmetic coder. The results are shown in Fig. 5 (a). We see that the DS methods is very costly in terms of additional rate, while the CS and DC strategies leads to a reasonable rate. We compare these results with a widely adopted solution that consists in transmitting a second reference view (which is encoded using JMVM standards [14]), so that the decoder receives Y_1 and Y_2 with generally $\bigcup_{X\in\mathcal{X}(Y)} \subset S_{Y_1} \cup S_{Y_2}$. The obtained results are very interesting since our approach requires less bit rate than the usual solution that jointly encodes two reference views. It shows that our approach may be adopted for multiview coding also. The results obtained on the *rect 1* amd *rect 2* sequences are further confirmed by experiments on natural sequences. We show in Fig. 6 and 5 (b) an example of what we obtain for *ballet* sequence. Similar results were observed on other videos such as *breakdancer*.

We finally propose an approach to measure the video quality at the decoder. The reconstruction of an image X is denoted by $\hat{X}$. The quality of the reconstruction is usually measured by mean-square error: $d(X,\hat{X}) = \frac{1}{N}\sum_{i=1}^{N} E(||x_i - \hat{x}_i||^2)$. This measure is often debatted [15] because it does not reflect the real visual quality of the images, since it does not take into account the flickering artifacts. Moreover, in virtual view synthesis, the original frames generally do not exist. We consider here an alternative way of measuring quality. Instead of considering the image domain, we propose to calculate the distortion in the scene domain. Secondly, instead of considering the



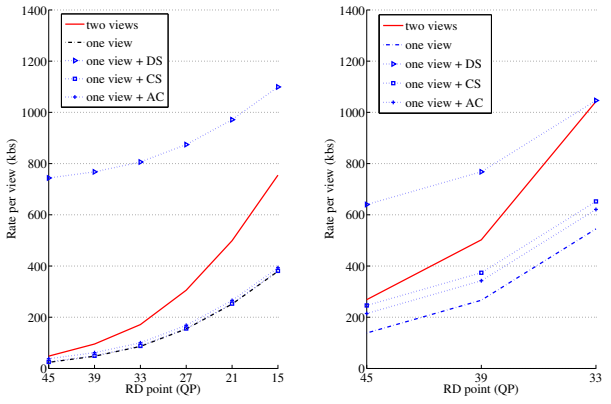
(a) No auxiliary information

(b) AI = Downsampling (DS)

(c) AI = Check sum bits (CS)

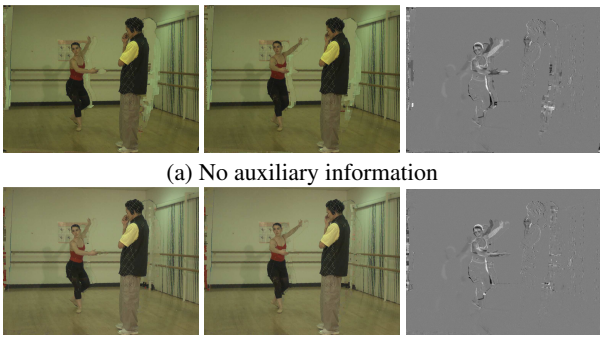
(d) AI = quantized DC coefficient (DC)

Fig. 4. Example of synthesized views generated with view 1. First column: reconstructed image at t=1. Second column: reconstructed image at t=2. Third column: image difference between the two consecutive reconstructed frames.



(a) *rect 1* (b) *ballet*

Fig. 5. Rate comparison at different RD points



(a) No auxiliary information

(b) AI = quantized DC coefficient (DC)

Fig. 6. Example of synthesized views generated with view 1 of *ballet*. First column: reconstructed image at t=1. Second column: reconstructed image at t=2. Third column: image difference between the two consecutive reconstructed frames.

fidelity with respect to an original frame, we estimate the coherence of the reconstruction with the neighboring images (in view and time). If $X_{t,n}$ corresponds to the frame of view n at time instant t, we define the incoherence γ as:

$$\gamma = \sum_{t'=t-1}^{t+1} \sum_{n'=n-1}^{n+1} E_{S_{X_{t,n}} \cap S_{X_{t',n'}}} (||f_{X_{t,n}}(x_i) - f_{X_{t',n'}}(x_i')||^2).$$

This metric constitutes the intuitive extension of well- accepted measures that make the difference between consecutive frames in order to quantify the temporal consistency [16]. We show the corresponding results in Tab. 1. We observe that the DS and DC methods significantly increase the coherence between the frames with respect to the case where no AI is sent to the decoder. The improvement is less significant on natural images since they present smoother content. In that case, the inconsistency is also due to the imprecision of the depth which is another problem. DC method, with an improvement of 0.4 w.r.t. the case of no AI transmission, brings however 50% of the possible improvement between the case of no AI and two views transmission. The proposed approaches do not constitute the optimal coding solution of $\mathbf{\Phi}$ but some of them (especially DC which achieves good quality for a low additional cost) offers promising perspectives in novel information representation methods for interactive multiview imaging.

	two views	no AI	DS	CS	DC
rect 1	0.0	5.0	2.3	5.3	1.2
rect 2	0.0	3.6	1.24	3.6	1.8
ballet	2.3	3.1	2.5	3.1	2.7

Table 1. Incoherence evaluation.

5. CONCLUSION

In this work, we propose a new multiview representation that complete the classical color and depth information streams with an auxiliary information that roughly describes the occlusion region in order to help the synthesis algorithm at the receiver. The results presented in the paper show that our solution enables to generate good quality synthesized views with a very reasonable additional rate. Moreover, the comparison with JMVM shows that our approach is even more interesting than transmitting another reference view. Further work will be conducted to investigate deeper the promising potential of this approach.

6. REFERENCES

[1] K. Müller, P. Merkle, and T. Wiegand, "3d video representation using depth maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[2] C. Fehn, "Depth-image-based rendering (dibr), compression and transmission for a new approach on 3d-tv," *Proc. SPIE, Stereoscopic Image Process. Render.*, vol. 5291, pp. 93–104, 2004.

[3] I. Daribo and H Saito, "A novel inpainting-based layered depth video for 3d-tv," *IEEE Trans. on Broadcasting*, vol. 57, no. 2, pp. 533–541, Jun. 2011.

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

[5] JG. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," *Proc. ACM Multimedia*, pp. 161–170, 2005.

[6] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview video coding using 3-d warping with depth map," *IEEE Trans. on Circ. and Syst. for Video Technology*, vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[7] G. Cheung, A. Ortega, and NM. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *IEEE Trans. on Image Proc.*, vol. 3, no. 3, pp. 744–761, Mar. 2011.

[8] J. Pandel, "Measuring of ickering artifacts in predictive coded video sequences," in *Internat. Work. on Image Analysis for Multim. Interactive Services*, Klagenfurt, Austria, May 2008.

[9] M. Köppel, P. Ndjiki-Nya, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegland, "Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering," in *Proc. IEEE Int. Conf. on Image Processing*, Honk Kong, Sep. 2010.

[10] SB. Lee and YS. Ho, "View-consistent multiview depth estimation for 3d video generation," in *3D TV Conference*, Tempere, Finland, Jun. 2010.

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

[12] ITU-T and ISO/IEC JTC1, "Joint scalable video model jsvm-8.6," Tech. Rep., 2007.

[13] R. Gallager, "Low density parity check codes," *MA: MIT Press*, vol. 0, no. 0, 1963, Cambridge.

[14] ISO/IEC MPEG & ITU-T VCEG, "Joint multiview video model (JMVM)," Marrakech, Morocco, Jan.13-19 2007.

[15] B. Girod, "What's wrong with mean-squared error?," *Digital Images and Human Vision*, pp. 207–220, 1993.

[16] D. Min, S. Yea, and A Vetro, "Temporally consistent stereo matching using coherence function," in *3D TV Conference*, Tempere, FInland, 2010.