INTERACTIVE MULTIVIEW VIDEO SYSTEM WITH LOW DECODING COMPLEXITY

Thomas Maugey, Pascal Frossard

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

ABSTRACT

Research in multimedia is always investigating new ways of improving the immersive experience of the users. One current solution consists in designing systems which offer a high level of interactivity, such as multiview content navigation where the point of view can be changed while watching at a video sequence (e.g., free viewpoint television, gaming, etc.). The coding algorithm designed for the transmission of such media streams must be adapted to these novel decoder needs. However, video plus depth data transmission is usually performed by considering the information flows as two sequences encoded with MVC schemes. Whereas it achieves good compression performance, this coding approach is not appropriate for interactive applications since the decoding of a frame often requires the prior transmission and decoding of several reference frames. Moreover, the techniques recently developed to improve interactivity are generally implemented at the decoder, whose computational complexity requirements are augmented. In this paper, we propose a novel coding scheme for video plus depth sequences that is adapted to user navigation; contrarily to several common approaches, the additional complexity is added on the encoder side so that the decoder stays simple. We further propose to limit the additional bandwidth imposed by interactivity requirements by designing a rate allocation algorithm that builds on a model of the user behavior. A first version of our novel coding architecture is evaluated in terms of rate-distortion performance, where it is shown to offer a high interactivity at a reasonable bandwidth cost.

Index Terms— 3D video coding, depth, interactive navigation

1. INTRODUCTION

More and more multimedia applications are intended at offering a 3D effect on the user side. This is not only achievable by using stereoscopic systems [1] that require specific datas and specific hardwares at the decoder. Indeed, if a user can change in real-time his point of view, he gets an impression of immersion or augmented visual content. That is why allowing interactivity in multiview system is a challenging but important problem that has many applications such as free viewpoint television or gaming.

Multiview sequences are usually represented as color plus depth sequences that are compressed with the JMVM algorithm [2]. Such solution however does not fit the interactivity requirements as it creates too many coding dependencies in the compressed stream. It is not adapted to rapid change of viewpoints, as it often requires the delivery and decoding of multiple reference frames prior to the display of the selected view. Techniques have been proposed in the literature to tackle interactivity in multiview content, especially for the application of free viewpoint television [3]. They however handle user interactivity by modifying the decoder [4, 5]. These approaches require additional computational power at the decoder that has to construct synthetic views and to fill in occluded regions with relatively complex methods [6]. Some works consider the interactivity at the encoder [7][8]. They allow the user to switch between the existing views. The encoder anticipates the user behavior by storing different predicted versions of the frames depending on the past reference image or by using distributed source coding techniques to reduce the amount of data at the server [8]. However, these schemes only permit the display of the views that have been captured and are rapidly limited when the number of cameras increases, although the number of views can become larger than > 50 in more and more situations [9]. Moreover the iterative algorithms performed by a distributed source decoder are complex.

In this paper we propose a new coding architecture for compressing multiview sequences (*i.e.*, N cameras (or N views) capturing the same scene from different viewpoints) while supporting user navigation with no additional complexity at decoder. We assume that the decoder is able to decode 2D plus depth content; the encoder prepares the multiview content along with intermediate views in order to offer smooth view switching capabilities at the decoder. Most of the navigation freedom is thus given by the encoder; it permits to change viewpoints in realtime with a reasonable rate penalty. We further propose a rate allocation algorithm that builds on a model of the user behavior in order to trade-off the quality of the images on the navigation path and the rate requirements. In addition, the bit rate can also be controlled by adapting the reactivity of the system and the smoothness of the view switching process. We evaluate our novel coding solution and shows that full interactivity can be offered with a reasonable rate penalty, even if our coding strategy is not fully optimized yet. In addition, the rate allocation algorithm provides important benefits in terms of rate-distortion performance without significant impact on the quality in the content navigation. The performance of our system highlights the compromise between rate-distortion performance and flexibility of navigability of the content in the design of solutions that offer increased interactivity.

2. MULTIVIEW CODING SCHEME

The initial set of camera is composed by N color+depth sequences. From this set, the encoder generates M virtual views, where the synthetic views situated between two cameras are disposed so that they form a linear path between two original views. With this set of $N + M$ views, the user has the ability to navigate to the left or to the right of the current view; he can further stay on each of the views (synthetic or original). Due to the network latency that is denoted by N_T, the server cannot be informed immediately of the change of view at the decoder. We assume that feedbacks are sent periodically every N_T inter-frame intervals. The encoder thus needs to anticipate the user navigation for the next N_T frames and prepare content that permits navigation during this period. In other words, when the user displays a view at a time t, the server sends the whole set of achievable frames during the next N_T time instants, in order to offer a complete interactivity during this period. At the end of each period, the same process is repeated and adapted to the past navigation pattern of the user.

From the point of view of the encoder, the sequence is not considered anymore as a set of N reference views, which are used at the decoder to generate M synthesized views, but it is directly seen as a set of $N + M$ views. The sequence is split into two types of frames. The first type belongs to the reference view and the second type of frame belongs to the synthesized views. The structure of the GOP and the two types of frames are shown in Fig. 1.

The reference views (color and depth) are coded with the standard H264 codec [2]. For the sake of simplicity, the interval between intra-coded frames (e.g., GOP) is aligned with the value of the period N_T in the proposed coding scheme. However, with small N_T values, it is possible to work with larger GOP sizes in order to enhance the reference view coding efficiency.

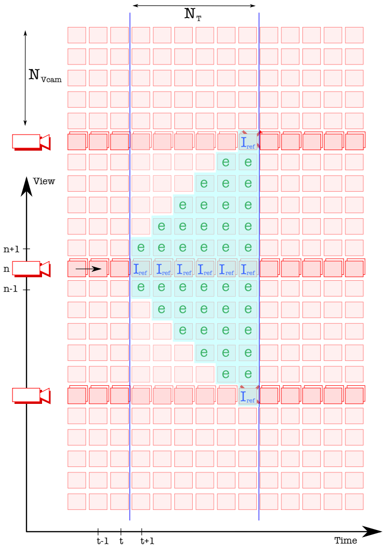

Fig. 1. Example of a GOP structure. With the current user navigation represented by the arrow, the achievable frames are in light blue.

The synthetic views are composed by the so-called *e frames*. As represented in Fig. 1, the server only encodes and transmits the achievable e frames given the current selection of the user. For every achievable e frame at a position (t_e, n_e), *i.e.*, n_e^{th} view and time t_e, the encoder first constructs an estimation of the synthetic view by warping the corresponding reference frame (at the position (t_e, n_{ref})), using the corresponding depth information and camera parameters. Then, the encoder calculates a residual for the missing regions due to occlusions. In our system, we have simply calculated the residual by taking the pixels of the original synthesized view (see Section 4) in the occluded regions. However, this residual estimation can also be done with classical inpainting methods [10] or with more advanced techniques like occlusion filling in algorithms [11]. Then, the encoder compresses the residuals and transmits them to the receiver. Note that the coding technique used for the e frames is independent from the rest of the scheme and may be adapted to the capabilities of the decoder. For instance one can consider algorithms inspired by the distributed source coding approach or even from predictive coding. In our current system, we use a solution that consists in encoding these residuals with an H.264 intra codec. Although it may lead to suboptimal RD performance, it offers a high level of flexibility. When a coder uses long prediction paths for compression one might need to transmit and decode several intermediary frames even if they are actually not displayed. We want to avoid this problem by adopting an intra coding approach. Hence, the decoder can directly access and decode the frame of interest.

3. RATE ALLOCATION WITH BEHAVIOR MODEL

The coding scheme presented above is blind to the user behavior and may lead to suboptimal rate-distortion performance. It only tries to offer a maximum of flexibility but does not consider any specific model of navigation in order to enhance its coding efficiency. In this section we propose a rate allocation based on a user behavior model. The expected rate-distortion performance is thus improved by considering the probability of choosing the different views on a navigation path, similarly to [8] that however uses a very simple probability model.

The general idea of the navigation model is based on prediction of the user behavior. The probability distributions of the different views can also be influenced by the popularity of the views or by the visual content itself. For example, if the user is already switching from left to right (or from right to left), he would unlikely come back to the previous view. He would most probably stay on the new viewpoint or continue moving in the same direction. Also, we might consider that a user is more likely to stay on the current than to switch to another view. Such simple observations about the user navigation can be stated in terms of transition probabilities that depends on an initial state of the user behavior.

More formally, we assume that the current displayed frame has the index (t, n), *i.e.*, time t, view n, and the one displayed just before had the index $(t-1, n^-)$, where $n^- \in \{n-1, n, n+1\}$[1]. A navigation model consists in estimating the probabilities of appearance, $p(n^+|n, n^-)$, for the next frames $(t+1, n^+)$, knowing the past displayed frames, (t, n) and $(t-1, n^-)$. The popularity, $P(t, n)$, of a frame, *i.e.*, the probability of appearance of the frame (t, n) is calculated recursively by adding the probability of all the possible navigation path which reaches (t, n).

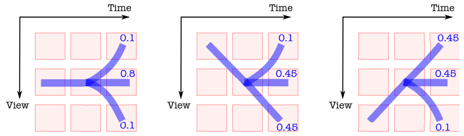

Fig. 2. Sample user navigation transition probabilities for 3 cases where the user either stays on the same views or switches to a neighboring view.

With a model of the user behavior, the encoder can calculate the frame appearance probabilities at each time t in the current GOP. The encoding performance can then be improved by the allocation of more bit rate to the frames that have higher chance to be displayed by the user. Based on these probabilities $P(t, n)$, the encoder implements a rate allocation algorithm that adapts the quantization of the residuals to these probabilities. In other words, the encoder solves a problem of the form

$$\min_{\mathbf{r}} \sum_n \sum_t D(\mathbf{r}(t,n))P(t,n) + \lambda||\mathbf{r}||_1$$

[1] Note that if the frame rate decreases, the number of intermediary views should decrease also (for bandwidth reason), and then, the set of achievable view $\{n-1, n, n+1\}$ remains the same.

where $\mathbf{r}$ is the rate distribution vector and $D(\mathbf{r}(t,n))$ is the distortion of the frame at instant t in view n, encoded with the rate $\mathbf{r}(t,n)$. In our system, we simply adapt the quantization parameter in the intra-mode encoding of the e frames following a linear function of the probability $P(t,n)$. Note that more complex models could be used for solving the above relation with better accuracy. Such models would however not modify dramatically the performance of the rate allocation strategy and we will leave them for future study.

Finally, we present briefly the behavior model that is used in this paper. We can notice however that the allocation algorithm is generic and can work with any navigation model that provides the frame appearance probabilities at a time t. Based on the observations above we have defined a model with the following probabilities that are graphically represented in Fig. 2:

$$\begin{aligned}
p(n+1|n,n-1) &= p(n-1|n,n+1) = p(n|n,n-1)\\
&= p(n|n,n+1) = 0.45\\
p(n+1|n,n+1) &= p(n-1|n,n-1) = p(n-1|n,n)\\
&= p(n+1|n,n) = 0.1\\
p(n|n,n) &= 0.8\\
p(n+1|n,n+1) &= p(n-1|n,n-1) = 0.1
\end{aligned}$$

The frame appearance probabilities can be computed recursively in frame group. Fig. 3 gives an illustration of the resulting probabilities values with our sample user navigation model given by the transition probabilities in Fig. 2

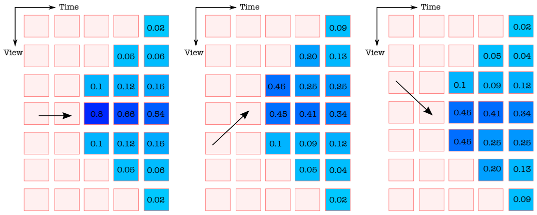

Fig. 3. Sample distribution of the probabilities of appearance for each e frame at each time t.

4. EXPERIMENTAL RESULTS

We evaluate here our multiview encoding algorithm in terms of rate-distortion (RD) performance. We have first created a groundtruth version of the synthetic frames. For this purpose we have first generated two versions of the estimated view (with the two closest neighboring reference views and the depth map). Then we have filled in the remaining holes with classical inpainting methods [10]. The tests have been run with the 50 first frames of two non registered multiview plus depth video sequences, *ballet* and *breakdancer*, both with a resolution of 768 × 1024, 30 frames per second (fps). Based on 8 equidistant reference video sequences, we have obtained 71 different views by generating 9 virtual views between each pair of neighboring cameras. We compute the value of the coding rate in our algorithm as the sum of the rates of the reference frames and the e-frames. The distortion corresponds to the weighted average distortion measured on the whole GOPs, with weights corresponding to the frames popularities. We assume here that the user model is well known and thus is the same for allocation and experiments. The model can obviously evolved without changing the general structure of the proposed scheme that only needs the frame popularities.

	Total rate	Color rate	Depth rate	e frame rate
GOP 4	22.59	16.49 (73 %)	3.21 (14 %)	2.89 (13 %)
	6.99	4.26 (61 %)	1.68 (24 %)	1.05 (15 %)
	2.83	1.51(53 %)	0.80 (28 %)	0.52 (19 %)
	1.57	0.78 (49 %)	0.37 (24 %)	0.42 (27 %)
GOP 8	16.39	11.61 (71 %)	2.47 (15 %)	2.30 (14 %)
	5.14	2.84 (55 %)	1.28 (25 %)	1.01 (20 %)
	2.37	1.00 (42 %)	0.59 (25 %)	0.78 (33 %)
	1.53	0.50 (33 %)	0.26 (17 %)	0.76 (50 %)

Table 1. Coding rates for each type of frames, *breakdancer* [in Mbs], for GOP size of 4 and 8, and for 4 bit rates.

	Total rate	Color rate	Depth rate	e frame rate
GOP 4	23.38	7.27 (31 %)	3.51 (15 %)	12.59 (54 %)
	8.71	2.22 (26 %)	2.04 (23 %)	4.45 (51 %)
	3.81	1.07 (28 %)	1.13 (30 %)	1.61 (42 %)
	2.11	0.59 (28 %)	0.57 (27 %)	0.96 (45 %)
GOP 8	16.20	4.21 (26 %)	2.51 (15 %)	9.48 (59 %)
	5.23	1.16 (22 %)	1.43 (27 %)	2.63 (50 %)
	3.05	0.55 (18 %)	0.78 (26 %)	1.71 (56 %)
	2.35	0.29 (13 %)	0.38 (16 %)	1.68 (71 %)

Table 2. Coding rates for each type of frames, *ballet* [in Mbs], for GOP size of 4 and 8, and for 4 bit rates.

The first results describe the general behavior of the proposed coding scheme. For different sizes of GOP (4 and 8) and for four quantization points we have measured the rates allocated for the reference views, the depth images and the e frames. For these experiments the quantization parameters (between reference camera and e frames) are manually set so that the average distortions of the two sets of images are similar. The results are presented in Tab. 1 and 2 for both sequences, respectively. One can remark that the distribution of the total bit rate clearly depends on the video sequence. Indeed, for *breakdancer* sequence, the e frames seem to be more efficiently compressible than the one of the *ballet* sequence. The percentages are thus much different, and this can be explained by several factors such as camera distribution, quality of the depth maps or the complexity of the scene. The percentage of the rate for the e frames might seem relatively high a priori. First, if we express them in terms of rate per frame, the e frames appear as sensibly lighter, since the number of frames in virtual views is actually pretty high. However, one has to keep in mind that the presented results correspond to fully flexible navigation, where the user can change the view at every time. The number of e frames can be reduced in order to control the additional rate and work on a trade-off between necessary bandwidth and visual navigation quality. Finally, the e frame coding is performed with intra coder for more interactivity and less time latency (low number of intermediary frames), but we can surely improve their compression efficiency by more efficient coding.

In order to analyze another aspect of the coder, we have compared the RD performances for different sizes of GOP (4 and 8). On the one hand, for a small GOP size the number of e frames is increased as well as the corresponding coding rate (Tab. 1 and 2); on the other hand, a large GOP size makes the inter coding of the reference depth views more efficient, as it can be seen in Tab. 1 and 2. At the end, if we measure the ΔPSNR in dB with the Bjontegaard met-

	GOP 4	GOP 8
breakdancer	0.29 dB	0.07 dB
ballet	0.36 dB	0.38 dB

Table 3. ΔPSNR (Bjontegaard metric [12]) between the schemes with no rate allocation and with model-based rate allocation.

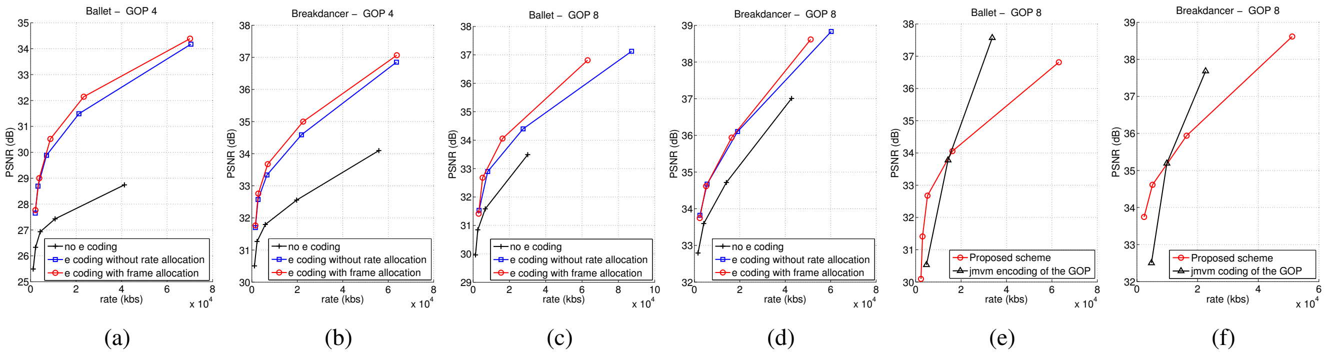

Fig. 4. Rate-distortion performance evaluation. Comparison between: (a-d) the three coding schemes under consideration: no e frame Coding, e frame coding with no rate allocation and e frame coding with the proposed allocation, (e,f) the proposed coding scheme and the JMVM coding applied to complete GOPs.

ric (commonly used for measuring the average difference of PSNR (dB) between two RD curves [12]), between the scheme with a GOP size of 4 and with a GOP size of 8, the higher GOP size brings an improvement of 1.4 dB for *breakdancer* and of 2.6 dB for *ballet*.

In terms of interactivity, the proposed approach provides the user with realtime navigation in the multiview content, with smooth transitions between views due to the presence of virtual camera views. This is due to the general structure of the scheme and to the intra coding approach for the e frames. We try now to compare our approach to other solutions for the same level of interactivity. As we are not aware of any baseline scheme in the literature, we build a simple but rather intuitive solution that does not use e frames and implements a simple interpolation strategy a decoder, where the missing information is replaced by a constant value that depends on the available information at neighboring pixel positions. The performance of this baseline scheme is reported by the black curves in Fig. 4 (a-d). The comparison with our proposed solution unequivocally shows that the proposed scheme in red is largely more efficient. In other words, the e frames bring important information for the visual quality at the decoder side. Another important test consists in measuring the efficiency of the rate allocation method based on the user behavior model. One can see in Fig. 4 (a-d) that the scheme with optimized rate allocation outperforms the one whose rate allocation does not consider any particular behavior model. As it can be seen in Tab. 3, the gain is quite small for *breakdancer* but becomes more sensible for *ballet*. This is again explained by the fact that the e frame residuals for *breakdancer* have smaller energy than in *ballet*.

Finally, another approach consists in encoding/decoding the depth and texture sequences for the whole GOP with a MVC codec [2], at the price of low flexibility for efficient navigation due to the presence of long coding dependency paths. The results are presented in Fig. 4 (e,f). One can see that for low bit rates, our scheme outperforms the JMVM performance since the proposed approach does not transmit any motion vectors for the virtual cameras. However, at high bit rates, it becomes less efficient. This is due to the intra coding of the e frames, which is less performant (but again more flexible) than a scheme based on prediction such as JMVM.

5. CONCLUSION

In this paper we have proposed a novel coding scheme that offers interactivity to the user by keeping a low decoding complexity. The preliminary results with this initial implementation are promising in terms of rate-distortion performance since interactivity can be achieved with a reasonable rate penalty. Moreover we have shown that the encoder should take into account the user navigation behavior for the rate allocation, in order to improve the compression efficiency without affecting interactivity. In the future we should improve the e frame coding by studying the trade-off between decoding flexibility and the RD performances and also build better models for user interaction and analyze the effect on the performance. Moreover it should be interested to measure the benefits of using two reference views for frame synthesis.

6. REFERENCES

[1] P. Merkle, H. Brust, K. Müller, and T. Wiegand, “Stereo video compression for mobile 3d services,” in *3D TV Conference*, Cancun, Mexico, Jun. 2009.

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

[3] M. Tanimoto, MP Tehrani, T. Fujii, and T Yendo, “Free-viewpoint TV,” *IEEE Signal Processing Magazine*, vol. 11, pp. 67–76, 2011.

[4] A. Smolic and P. Kauff, “Interactive 3D video representation and coding technologies,” *Proc. IEEE*, vol. 93, pp. 98–110, 2005.

[5] C. Weigel, S. Schwarz, T. Korn, and M. Wallebohr, “Interactive free viewpoint video from multiple stereo,” in *3D TV Conference*, Postdam, Germany, May 2009.

[6] KJ Oh, S. Yea, and YS Ho, “Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3-D video,” in *Picture Coding Symposium (PCS)*, Chicago, IL, USA, May 2009.

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

[8] G. Cheung, A. Ortega, and NM Cheung, “Interactive streaming of stored multiview video using redundant frame structures,” *IEEE Trans. on Image Proc.*, vol. 3, pp. 744–761, 2011.

[9] K. M uller, P. Merkle, and T. Wiegand, “3d video representation using depth maps,” *Proc. IEEE*, vol. 99, pp. 643–656, 2011.

[10] 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.

[11] 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. Int. Conf. on Image Processing*, Honk Kong, Sep. 2010.

[12] G. Bjontegaard, “Calculation of average PSNR differences between RD curves,” Tech. Rep., 13th VCEG-M33 Meeting, Austin, TX, USA, Apr. 2001.