WYNER-ZIV CODING FOR DEPTH MAPS IN MULTIVIEW VIDEO-PLUS-DEPTH

Giovanni Petrazzuoli, Marco Cagnazzo, Frederic Dufaux, Béatrice Pesquet-Popescu

TELECOM-ParisTech, TSI department
46 rue Barrault, F-75634 Paris Cedex 13, FRANCE

ABSTRACT

Three dimensional digital video services are gathering a lot of attention in recent years, thanks to the introduction of new and efficient acquisition and rendering devices. In particular, 3D video is often represented by a single view and a so called depth map, which gives information about the distance between the point of view and the objects. This representation can be extended to multiple views, each with its own depth map.

Efficient compression of this kind of data is of course a very important topic in sight of a massive deployment of services such as 3D-TV and FTV (free viewpoint TV). In this paper we consider the application of distributed coding techniques to the coding of depth maps, in order to reduce the complexity of single view or multi view encoders and to enhance interactive multiview video streaming. We start from state-of-the-art distributed video coding techniques and we improve them by using high order motion interpolation and by exploiting texture motion information to encode the depth maps. The experiments reported here show that the proposed method achieves a rate reduction up to 11.06% compared to state-of-the-art distributed video coding technique.

Index Terms— Distributed video coding, image interpolation, multiview video-plus-depth, depth map coding

1. INTRODUCTION

Multiview video is gathering increasing attention, thanks to the availability of acquisition and rendering systems, paving the way to a large number of interesting research topics and applications, such as 3D and free viewpoint TV (FTV) [1]. In this context, the problem of efficient compression is more urgent than ever, in sight of the huge amount of data required to represent 3D video. In particular, a representation which is gaining attention from the research and industry communities is based on the use of depth map in addition to a standard texture. This allows an easy generation of more virtual views with depth-image-based rendering (DIBR) methods [2]. Moreover, systems with multiple views, each one with its depth map are already envisaged, since they promise a simple implementation of 3D and FTV [3]. This approach is called multiview video-plus depth (MVD). In this context, each camera produces a regular, color image, accompanied by a 8-bit depth map, which is obtained via infrared sensors (see Fig. 1 for an example).

Two very important issues in MVD are coding and transmission. In this work we focus only on the first one, which is usually faced by coding separately the color video and the depth data by a simulcast coding structure (where each view is coded independently) or a multiview-coding structure (where the views are jointly coded) [4]. For example Ho and Oh [5] share the motion information between the texture and the depth. They proposed to code the texture video with H.264/AVC. For the depth maps, instead of estimating the vectors directly on them, candidate motion modes are generated by exploiting motion information of the corresponding texture video.

Fig. 1. At each instant the output of each Z-camera is the color image and the corresponding 8-bit depth map

An extension of H.264/AVC to multiview video coding (MVC) has been proposed [6] in order to exploit also the inter-view correlation. For the depth map, instead of estimating motion vectors directly on the depth map, candidate motion modes are generated by exploiting motion information of the corresponding texture video.

In this paper we consider a DVC approach [7] to the MVD problem. This approach makes sense in many application scenarios: for example, acquisition systems could prevent inter-camera communication; or, one may be interested in low-complexity encoding techniques which do not make use of motion or disparity estimation/compensation (ME/MC or DE/DC).

In particular, we consider one of the most popular distributed video coding systems, that is the one proposed by Girod et al. [7]. In this framework, a video stream is structured into groups of pictures (GOP), in which some selected frames, called key frames (KF), are coded using any "intra" method. Usually, KFs are coded with the Intra mode of H.264/AVC with an assigned quantization parameter (QP). For the other frames, called Wyner-Ziv frames (WZFs), only the parity bits, obtained by a channel encoder, are sent to the decoder. These bits are used to correct an initial estimation of the WZF obtained from the decoded KFs. This estimation is called side information (SI) and the process generating it is called image interpolation. The image interpolation step has a crucial importance in a DVC system, since the number of bits needed to correct the SI decreases as its quality increases. We remark that we can modify the image interpolator without affecting the encoder.

The DVC approach can easily be extended to the MVD case, as shown in Fig. 2. In this example, each view is composed by a sequence of texture images and another sequence of depth maps. Each sequence is DVC-coded, and this for each view. The advantage of the presence of the WZF in the video stream is the flexibility of this structure in the context of interactive multiview video [8]. In fact, the WZF can be decoded independently from the reference frame available at the decoder. If the video is coded with H.264/AVC and

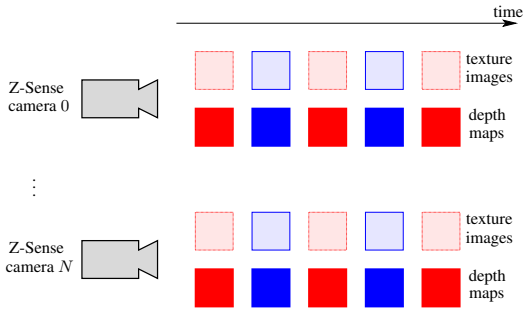

Fig. 2. The frame repartition in time-view domain used in this paper (GOP size equal to 2). The KFs are in red, while the WZs are in blue. The texture images and the depth maps are independently coded.

if the switching point corresponds to a P-frame/B-frame, the latter can not be decoded because the reference frame is not available at the decoder. This problem does not exist if the P-frames/B-frames are replaced by WZFs.

In this paper we propose a method to improve the quality of the side information for the depth maps. We focus on the case when one view is coded independently from the other. The basic idea is inspired by the high order motion interpolation method that we proposed previously for single view or multiview videos without depth information [9]. In this paper this method is modified in order to take into account the specificity of depth maps, and in particular to take advantage of the motion information coming from the texture images: the vectors for the depth maps are not estimated directly on them, but the corresponding vectors on texture video are used and they are refined on the depth maps. In this way we can improve upon state-of-the-art DVC techniques for the depth map coding, as reported in the experimental section.

The rest of the paper is organized as follows. In section 2, the state-of-the-art method used to initialize our algorithm is described. Then, in section 3 we give the details of our technique. Experimental results are reported in section 4, while section 5 ends the paper and outlines future work.

2. APPLICATION OF DISTRIBUTED SOURCE CODING TO MULTIVIEW VIDEO-PLUS-DEPTH

Usually the depth maps are encoded independently of the texture images by a simulcast coding structure, where the depth maps of each view are coded independently or by a multiview-coding structure, where the depth maps of different views are coded jointly [4]. Note that in those schemes the depth maps are coded by a standard coding technique (e.g. H.264/AVC). We propose to apply DVC to the encoding of the depth maps: the texture images and the depth maps are independently coded using DISCOVER [10]. The frames are split into KFs and WZFs, as illustrated in Fig. 2.

We describe the reference technique for image interpolation, which is the one implemented in DISCOVER [10]. It can be seen as a linear motion interpolation between two adjacent KFs. For example, if the GOP size is equal to 2, we consider as KFs the images I_{k-1} and I_{k+1} [1]. Our goal is to determine an estimation of the WZF I_k, by using I_{k-1} and I_{k+1}. In particular, we consider a macroblock (MB) of I_k centered in $\mathbf{p}$.

[1]The frame I can be indifferently the texture image or the depth map

At first, the two KFs are spatially filtered to reduce noise. We search for the vector $\mathbf{v}$ that intercepts the frame k in the point closest to $\mathbf{p}$. Let $\mathbf{q}$ be this point.

Then, we split the vector $\mathbf{v}(\mathbf{q})$ in

$$\mathbf{u}(\mathbf{p}) \triangleq \frac{1}{2}\mathbf{v}(\mathbf{q}) \qquad \mathbf{w}(\mathbf{p}) \triangleq -\mathbf{u}(\mathbf{p})$$

and we center them in $\mathbf{p}$. The vectors $\mathbf{u}(\mathbf{p})$ and $\mathbf{w}(\mathbf{p})$ are refined around the positions $\mathbf{p}+\mathbf{u}(\mathbf{p})$ and $\mathbf{p}+\mathbf{w}(\mathbf{p})$, using a vector $\mathbf{e} \in W = \{-1,0,1\} \times \{-1,0,1\}$: we search for $\mathbf{e}$ that gives the best matching between the macroblocks $B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})+\mathbf{e}}$ and $B_{k+1}^{\mathbf{p}+\mathbf{w}(\mathbf{p})-\mathbf{e}}$, where we define $B_\ell^{\mathbf{p}}$ as the MB of the frame I_ℓ centered in $\mathbf{p}$. Then, the two new vectors are

$$\mathbf{v}^{\mathrm{NEW}}(\mathbf{p}) \triangleq \mathbf{v}(\mathbf{p}) + \mathbf{e} \qquad \mathbf{w}^{\mathrm{NEW}}(\mathbf{p}) \triangleq -\mathbf{v}^{\mathrm{NEW}}(\mathbf{p})$$

Afterwards, a weighted median filter is run over the motion vectors in order to regularize them. The vectors computed in this way are used for compensating the KFs and the average of the resulting images constitutes the side information.

We consider the DISCOVER technique used on depth maps as our reference. In fact, it is one of the most popular techniques in DVC coding. In order to distinguish it from the proposed methods, it is labeled as "ZD", that is DISCOVER on the Z (or depth) data.

3. PROPOSED METHOD: HIGH ORDER MOTION INTERPOLATION (HOMI)

The algorithm for image interpolation in DISCOVER is computationally quite simple, but it carries out only a linear interpolation between adjacent key frames I_{k-1} and I_{k+1}. We propose to exploit also the key frames I_{k-3} and I_{k+3}, in order to perform a higher order interpolation and to increase the SI quality. In this way, we can include also acceleration of objects into our model. This method, called high order motion interpolation (HOMI), was proposed for texture image coding in our previous work [9], and we adapt it here to depth maps. For sake of clarity, we report here the basic ideas of HOMI (see Fig. 3). It consists in five steps:

1. The frames I_{k-3}, I_{k-1}, I_{k+1} and I_{k+3} are filtered in order to reduce noise
2. We estimate $\mathbf{u}(\mathbf{p})$ from I_k to I_{k-1} and $\mathbf{w}(\mathbf{p})$ from I_k to I_{k+1} for each macroblock by using DISCOVER, as described in the previous section.
3. We carry out block matching to find the position of $B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}$ in I_{k-3}. This position is called $\mathbf{p}+\widetilde{\mathbf{u}}(\mathbf{p})$. Likewise we define the vector $\widetilde{\mathbf{w}}$ by matching $B_{k+1}^{\mathbf{p}+\mathbf{w}(\mathbf{p})}$ in I_{k+3}.
4. By interpolating the positions $\mathbf{p}+\widetilde{\mathbf{u}}(\mathbf{p})$, $\mathbf{p}+\mathbf{u}(\mathbf{p})$, $\mathbf{p}+\mathbf{w}(\mathbf{p})$, and $\mathbf{p}+\widetilde{\mathbf{w}}(\mathbf{p})$ we obtain the new position $\widehat{\mathbf{p}}$
5. We define the new vectors as the displacement between $\widehat{\mathbf{p}}$ and $\mathbf{p}+\mathbf{u}(\mathbf{p})$ [resp. $\mathbf{p}+\mathbf{w}(\mathbf{p})$].

Some further details are needed to complete the description of the method. The block matching in the third step is performed around the positions obtained by extending vectors $\mathbf{u}(\mathbf{p})$ and $\mathbf{w}(\mathbf{p})$ in I_{k-3} and I_{k+3}, these are $\mathbf{p}+3\mathbf{u}(\mathbf{p})$ and $\mathbf{p}+3\mathbf{w}(\mathbf{p})$. Then, we search for the refinement vector $\delta\mathbf{u}$ [resp. $\delta\mathbf{w}$] such that the following functional is minimized:

$$J(\delta\mathbf{u}) = \sum_{\mathbf{q}} \left| B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}(\mathbf{q}) - B_{k-3}^{\mathbf{p}+3\mathbf{u}(\mathbf{p})+\delta\mathbf{u}}(\mathbf{q}) \right| + \lambda\|\delta\mathbf{u}\| \quad (1)$$

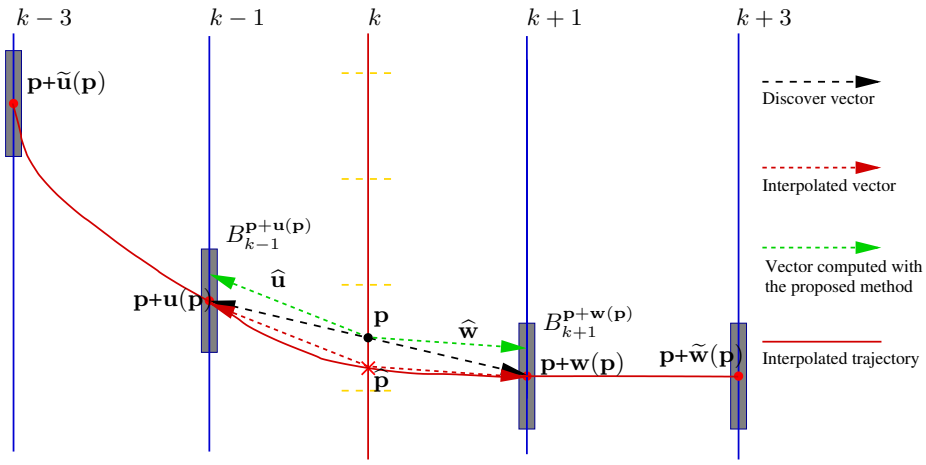

Fig. 3. Proposed interpolation method for motion estimation.

with $\lambda > 0$ a regularization constant. The regularization term penalizes a too large deviation from the linear model: with $\lambda \to \infty$ the proposed algorithm becomes equivalent to DISCOVER. Then, we define

$$\widetilde{\mathbf{u}} \triangleq 3\mathbf{u} + \delta\mathbf{u}$$
$$\widetilde{\mathbf{w}} \triangleq 3\mathbf{w} + \delta\mathbf{w}$$

The fourth step consists in interpolating the positions of the current block in the four images. By using a piecewise cubic Hermite interpolation we find the position $\widehat{\mathbf{p}}$ in the frame k. The interpolated motion vectors are shown in red in Fig. 3.

The last step consists evaluating the new motion vectors for $\mathbf{p}$ as:

$$\begin{aligned} \widehat{\mathbf{u}}(\mathbf{p}) &= \mathbf{u} + \mathbf{p} - \widehat{\mathbf{p}} \\ \widehat{\mathbf{w}}(\mathbf{p}) &= \mathbf{w} + \mathbf{p} - \widehat{\mathbf{p}} \end{aligned}$$

The HOMI algorithm can be used in order to find the motion vectors of depth maps using these images as key frames I_ℓ (with $\ell \in \{k-3, k-1, k+1, k+3\}$). We call this technique ZD-ZH, because we use the vectors obtained using DISCOVER on depth data, to initialize HOMI, which in turn is run over Z data. This technique has the potential to improve the results of DISCOVER for depth map coding. However, we want to better exploit the correlation between texture images and depth maps, and therefore to introduce other new coding methods.

We observe that motion in depth maps is very similar to motion in texture sequence. So we could use DISCOVER vectors computed on texture in order to perform the motion compensation and the image interpolation of depth maps. We call this method TD (DISCOVER over Texture).

This straightforward method can be improved if we refine TD vectors by using HOMI over texture data. We call the resulting technique TD-TH.

Finally we consider a last technique, which uses DISCOVER over texture data to initialize the HOMI algorithm, which on the other hand is run over depth-map data. This method is called TD-ZH. The rationale behind it is that texture data are much richer in information than depth maps, and it could provide an initialization that is closer to real object movement. However we point out that texture movement does not always correspond to depth map motion, and *vice versa*: for example, an object moving over a static background which is at the same distance from the camera, would not result in depth map movement. The proposed methods are resumed in Table 1.

Label	Initialization	Refinement
ZD	*DISCOVER on depth maps*	—
ZD-ZH	DISCOVER on depth maps	HOMI on depth maps
TD	DISCOVER on texture	—
TD-TH	DISCOVER on texture	HOMI on texture
TD-ZH	DISCOVER on texture	HOMI on depth maps

Table 1. Labels for reference (in italic) and proposed methods

4. EXPERIMENTAL RESULTS

We have implemented the reference method and all the proposed techniques shown in Table 1, and we have run several tests to validate and compare them to the reference. In a first stage, we use as evaluation metric the PNSR of the SI with respect the original WZF. More precisely for each of the new methods, we compute the PSNR difference with respect to DISCOVER (ZD). This quantity is called Δ_{PSNR}.

In a second stage, we compute end-to-end performances (*i.e.* rate reduction and PSNR improvement) of the proposed techniques when inserted into a complete DVC coder.

The experiments are conducted as follows. We use the test depth map sequences *ballet* and *breakdancer* at a resolution of 384×512 pixels. We encode the depth map KFs by the INTRA mode of H.264, using four quantization step values, namely 31, 34, 37 and 40. The Δ_{PSNR} values are computed as average along the sequences.

The optimal value of λ, *i.e.* the one minimizing the cost function $J(\cdot)$ in Eq. (1), is found from experiments. We observe that for the TD method λ is not needed, while for TD-TH we can use the values reported in our previous work for texture images [9]. We obtain the values shown in Table 2 by maximizing the average PSNR over all the sequences and at all the QPs. We observe that, when only texture data is used, we need a stronger regularization, while if they are used only as initialization λ must be smaller to enable larger correction (since vectors are not initialized on depth maps). However, if texture is not used at all, an intermediate regularization strength is needed.

By using the optimal parameters found in the previous section,

GOP size	2	4	8
$\lambda_{\mathrm{TD-TH}}$	50	20	0
$\lambda_{\mathrm{ZD-ZH}}$	10	8	2
$\lambda_{\mathrm{TD-ZH}}$	4	2	2

Table 2. Values of λ for different techniques and GOP sizes

ballet				
QP	ZD-ZH	TD	TD-TH	TD-ZH
GOP size = 2				
31	0.25	0.28	0.54	**0.56**
34	0.22	0.29	0.55	**0.57**
37	0.22	0.15	0.39	**0.41**
40	0.20	0.18	0.37	**0.41**
GOP size = 4				
31	0.34	0.15	0.56	**0.59**
34	0.25	0.04	0.49	**0.52**
37	0.32	0.00	0.49	**0.52**
40	0.27	-0.01	0.57	**0.62**
GOP size = 8				
31	0.34	0.06	0.29	**0.52**
34	0.31	0.05	0.28	**0.58**
37	0.27	-0.04	0.16	**0.54**
40	0.26	-0.04	0.14	**0.67**
breakdancers				
GOP size = 2				
31	0.11	0.06	0.10	**0.12**
34	0.11	0.06	0.11	**0.12**
37	**0.11**	-0.01	0.02	0.03
40	**0.09**	-0.03	0.00	0.01
GOP size = 4				
31	0.03	0.03	0.16	**0.17**
34	0.02	0.01	0.18	**0.18**
37	0.01	0.00	0.21	**0.22**
40	0.01	-0.02	**0.29**	**0.29**
GOP size = 8				
31	0.01	0.03	0.07	**0.16**
34	0.02	0.02	0.06	**0.19**
37	0.02	0.02	0.06	**0.25**
40	0.01	-0.03	0.03	**0.31**

Table 3. Δ_{PSNR} [dB] for sequence *ballet* and *breakdancers*

we compare the SI quality of all proposed methods with the ZD method. The complete results are in Table 3 and for the two sequences. We observe that almost all methods allow remarkable gains in SI quality, up to 0.6 dB. We notice that the TD-ZH method gives the best results. In fact, the initialization is better if computed on the texture image, because the depth data are too homogeneous and block matching can fail in finding the true movement. The refinement is computed on the depth map in order to have a better estimation.

Finally, we perform a complete end-to-end coding of video sequences. The rate-distortion performances, computed with the Bjontegaard metric [11], are shown in Table 4. We note that with the method TD-ZH we obtain a rate reduction up to 11.06% with respect to ZD and a PSNR improvements up to 0.44 dB.

5. CONCLUSIONS AND FUTURE WORK

In this paper we presented a set of new methods for depth map coding in the context of MVD and DVC, taking inspiration from our previous work [9]. We investigated how to model non linear motion and how to exploit the dependences between the texture image and the depth map at each instant. The reference technique for WZF estimation is DISCOVER. We show that the proposed methods allows remarkable PSNR improvements on the SI, up to 0.6 dB. Moreover, by using the proposed techniques in a complete DVC system, we can reduce the coding rate up to 11.06%. The use of distributed video coding allows to apply our algorithms in the context of interactive multiview video streaming. In fact, WZFs can be correctly decoded independently of which frames are available at the decoder. This may not be possible for P-frames/B-frames: when the user changes the view, the reference frame may not be available at the decoder side. The use of WZF allows to have a continue playback of the video during the view switching.

ballet				
	ZD-ZH	TD	TD-TH	TD-ZH
GOP size = 2				
Δ_{R} (%)	-3.46	-2.05	-5.83	**-6.08**
Δ_{PSNR} [dB]	0.17	0.10	0.29	**0.31**
GOP size = 4				
Δ_{R} (%)	1.22	8.29	-3.38	**-4.42**
Δ_{PSNR} [dB]	-0.04	-0.36	0.14	**0.17**
GOP size = 8				
Δ_{R} (%)	-4.34	7.22	-0.36	**-11.06**
Δ_{PSNR} [dB]	0.17	-0.29	0.04	**0.44**
breakdancers				
GOP size = 2				
Δ_{R} (%)	**-2.09**	1.51	-0.44	-0.93
Δ_{PSNR} [dB]	**0.09**	-0.07	0.01	0.04
GOP size = 4				
Δ_{R} (%)	-0.59	7.55	-0.55	**-0.94**
Δ_{PSNR} [dB]	0.02	-0.35	0.01	**0.02**
GOP size = 8				
Δ_{R} (%)	-0.96	7.27	3.66	**-4.50**
Δ_{PSNR} [dB]	0.03	-0.32	-0.16	**0.17**

Table 4. Rate-distortion performance for *ballet* and *breakdancers* by Bjontegaard metric [11]

6. REFERENCES

[1] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 67 –76, 2011.

[2] C. Fehn, "A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR)," in *Proceedings of 3rd IASTED Conference on Visualization, Imaging, and Image Processing*, Benalmádena, Spain, Sept. 2003, pp. 482–487.

[3] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, "An Evolutionary and Optimised Approach on 3D-TV," in *Proceedings of International Broadcast Conference*, Amsterdam, The Netherlands, Sept. 2002, pp. 357–365.

[4] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 16 2007.

[5] H. Oh and Y.S. Ho, "H.264-based depth map sequence coding using motion information of corresponding texture video," *Advanced Concepts for Intelligent Vision Systems*, vol. 4319, pp. 898–907, 2006.

[6] Ying Chen, Ye-Kui Wang, Kemal Ugur, Miska M. Hannuksela, Jani Lainema, and Moncef Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Appl. Signal Process.*, vol. 2009, pp. 8:1–8:13, January 2008.

[7] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[8] G. Cheung, A. Ortega, and N. M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures," *Image Processing, IEEE Transactions on*, 2010.

[9] G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, "High order motion interpolation for side information improvement in dvc," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 2342 –2345.

[10] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation, picture coding symposium," in *Coding of Audio-Visual Objects, Part 10: Advanced Video Coding, 1st Edition*, 2007.

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