Efficient Depth Map Compression based on Lossless Edge Coding and Diffusion

Josselin Gautier
IRISA / Université de Rennes 1
Campus de Beaulieu
Rennes, France 35042
Email: jgautier@irisa.fr

Olivier Le Meur
IRISA / Université de Rennes 1
Campus de Beaulieu
Rennes, France 35042
Email: olemeur@irisa.fr

Christine Guillemot
INRIA
Campus de Beaulieu
Rennes, France 35042
Email: christine.guillemot@inria.fr

***Abstract*— The multi-view plus depth video (MVD) format has recently been introduced for 3DTV and free-viewpoint video (FVV) scene rendering. Given one view (or several views) with its depth information, depth image-based rendering techniques have the ability to generate intermediate views. The MVD format however generates large volumes of data which need to be compressed for storage and transmission. This paper describes a new depth map encoding algorithm which aims at exploiting the intrinsic depth maps properties. Depth images indeed represent the scene surface and are characterized by areas of smoothly varying grey levels separated by sharp edges at the position of object boundaries. Preserving these characteristics is important to enable high quality view rendering at the receiver side. The proposed algorithm proceeds in three steps: the edges at object boundaries are first detected using a Sobel operator. The positions of the edges are encoded using the JBIG algorithm. The luminance values of the pixels along the edges are then encoded using an optimized path encoder. The decoder runs a fast diffusion-based inpainting algorithm which fills in the unknown pixels within the objects by starting from their boundaries. The performance of the algorithm is assessed against JPEG-2000 and HEVC, both in terms of PSNR of the depth maps versus rate as well as in terms of PSNR of the synthesized virtual views.**

I. Introduction

3DTV and FVV are promising technologies for the next generation of 3D entertainment. To this end, the MultiView plus Depth (MVD) format has been developed. From a collection of multi-view video sequences captured synchronously from different cameras at different locations, a view synthesis or rendering method can generate new viewpoints. The first version of the dedicated MVD sequence encoder, so-called MVC, was based on block transforms of depth maps. As argued by [5], the deblocking of depth maps is today one of the most important pre- and post-processing task for the MVD representation.

Depth maps have two main features that must be preserved but can also be relied on for efficient compression. The first one is the sharpness of edges, located at the border between object depths. Distortions on edges during the encoding step would cause highly visible degradations on the synthesized views, that may require depth map post-processing. The second one comes from the general smooth surface properties of objects we are measuring the depth on. Based on these observations, Merkle et al. [4] proposed a "Wedgelet" signal decomposition method (itself based on "Platelet"). The smooth regions of the depth maps are approximated using piecewise-linear functions separated by straight lines. A quadtree decomposition divides the image into variable-size blocks, each of them being approximated by a modeling function. The refinement of the quadtree is then optimized in the rate-distortion sense.

In this context, we also observed that these smooth surfaces can be efficiently approximated by interpolating the luminance values located at their boundaries, instead of using models based on piecewise-linear functions where the coefficients need to be encoded [4]. To this end, we can observe that depth maps share similarity to cartoon-images. Mainberger et al. [3] proposed a dedicated cartoon-image encoder, that -in low bitrate conditions- beats the JPEG-2000 standard. After a Canny edge detection, the edge locations are encoded with a lossless bi-level encoder, and the adjacent edge pixel values are lossy quantized and subsampled. At the decoding stage, a homogeneous diffusion is used to interpolate the inside unknown areas from lossy decoded edges. Indeed, the demonstrated performances -while beating state of the art codecs- reach the limit of 30dB. We revisited this edge-based compression method by proposing improvements to fit the high quality, low bitrate, and specific requirements of depth maps. Finally, we increase the diffusion-based depth map encoding performance, which might be generalized to all kinds of images. In the next section, the encoding process is described. In section III the new decoding and diffusion methods are explained. Results, performances and comparison with state-of-the-art methods are given in Section IV. Conclusions are then drawn in section V.

II. Encoding

The encoding is a 3-step process: first is the detection of edges, then encoding of the edge location and finally encoding of the edge, border and seed pixel values.

A. Edge detection

Different operators exist to extract the contour of an image. An optimal edge detector should provide:

- a good detection: the algorithm should find as much real edges as possible.

- a good localization: the edges should be marked as edges as close as possible to the real edges.
- a good robustness: as much as possible, the detector should be insensitive to noise.

In our context of depth map edge coding, several requirements are added. The quality of reconstruction by diffusion should be maximized, while minimizing the number of edges to encode. To avoid diffusion from bad positioned edges causing "leakages", the localization of contours should be quasi-perfect (see section III for explanation). The detection of contours should be good but avoiding an over-detection. Up to a certain limit, weak contours (i.e. with a low gradient) might be useless to the reconstruction and might unnecessarily increase the edge coding cost. Also, noisily detected pixels should be avoided for the same reason.

The Marr-Hildreth edge detector combined with Canny-like hysteresis thresholding is used in [3], but suffers from error of localization at curved edges. The widely used Canny edge detector has also been benchmarked. It relies on a 5x5 gradient prefiltering to cope with noise before local maxima edge detection. But this prefiltering step also makes this detector vulnerable to contour localization errors, as illustrated in Fig.1(c), where inexact selection of adjacent edge pixels lead to improper diffusion. At the opposite Sobel has the advantage of an accurate contour localization -as shown in Fig.1(d)- at the cost of a noisy, edge over-detection. To cope with these over-detected edges, contours c shorter than a certain value ($c < 14$) are excluded. Pixels with a bi-dimensional gradient amplitude larger than a threshold λ are extracted. Used with sharp depth maps, this gives well-localised contours.

B. Encoding the contour location

As in [3], a bi-level edge image containing the exact location of previously detected edges is first encoded using the *JBIG (Joint Bi-level Image Experts Group)* standard. This is a context-based arithmetic encoder enabling lossless compression of bi-level images. We use the JBIG-Kit [2], a free C implementation of the JBIG encoder and decoder. The progressive mode is disabled to reduce the required bitrate.

C. Encoding the contour values

Once the edge pixel locations have been encoded, the pixel luminance values have also to be losslessly encoded following our initial requirements. The authors in [3] proposed to store the pixel values on both sides of the edge, instead of the pixel values lying on the edge itself. Indeed, for blurry contours, this might be valuable to interpolate the inner part of the edge and code the luminance values on both sides. However, with sharp depth maps, the pixel values lying directly on an edge, as illustrated in Fig.1(b), alternate between one side or another from this edge and couldn't be interpolated correctly.

With the Sobel edge detection not thinned to a single edge pixel, we ensure to retain at least one pixel value from each side of the frontier edge as shown in Fig.1(d).

We keep the idea of storing the pixel values by their order of occurrence along the edge to minimize signal entropy. A path

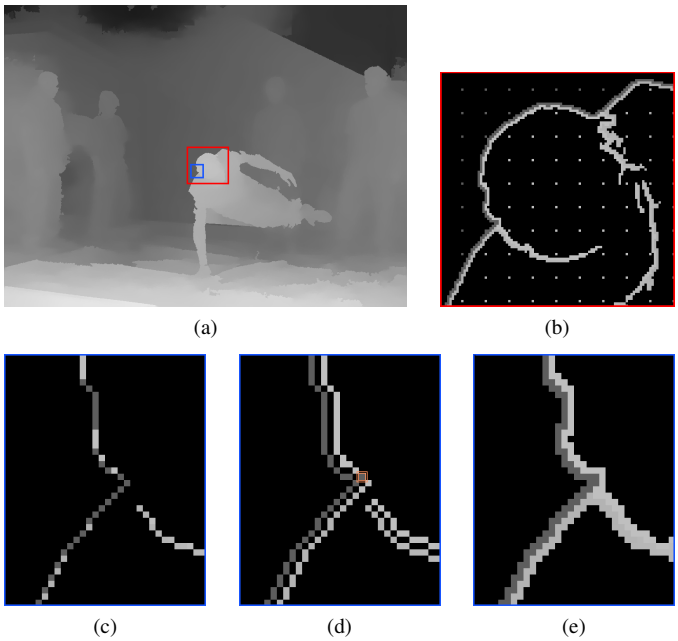

Fig. 1: (a) A "Breakdancer" depth map, (b) the encoded and decoded Sobel edge and seed pixels (red selection on (a)), (c) the Canny edges (blue selection), (d) the selection of pixel values adjacent to Canny edges (c) as in [3], with an intruder edge pixel in orange that will lead to bad diffusion, (e) the proposed Sobel selection of edge pixel values, exactly located from both side of the frontier edge

with fix directional priorities (E, S, W, N, NE, NE SE, SW and NW) is used. As the intrinsic properties of pixels along an edge or "isophote" are their small luminance variation, then we propose to compute the differential values of edge pixels in a Differential Pulse Code Modulation (DPCM) way. From this optimized path encoding method, the stream of DPCM values is then encoded with an arithmetic coder.

Additionally to these edges we also encode two kinds of information. The pixel values from the image border are stored to initiate the diffusion-based filling from borders. Inspired by the work of [1] on "dithering" for finding optimal data for interpolation, we propose to sparsely deploy, at regular intervals, some seeds of original depth pixels as shown in Fig.1(b). While having low overhead, we discovered that this helps accurate reconstruction by initializing and accelerating the diffusion in large missing areas.

Thus, these extra border and seed pixels are coded in DPCM and added to the differential edge values. The resulting file is thus composed of the payload of the JBIG data and of the arithmetic encoded bitstream of the DPCM edge, border, and seed pixel values.

III. Decoding and diffusion

A lossless decoding is performed, followed by a lossy diffusion from the decoded edges.

A. Decoding contour location and pixel values

Once the edge positions from JBIG payload are decoded, the edge pixel values are decoded and positioned following the same order in which they were encoded: the path along contour

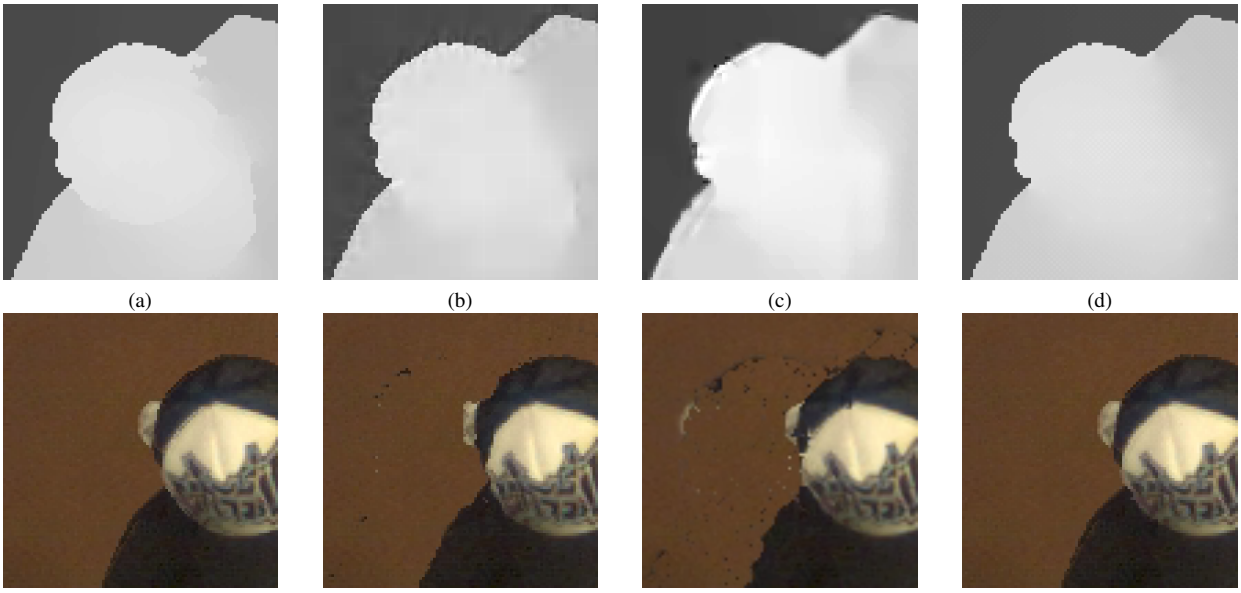

Fig. 2: Upper row: zoom on the head of a dancer on original View #3 (V_3) depth map (a) highlights -by comparison at equal depth map PSNR (45dB) referenced to (a)- the ringing artifact on JPEG-2000 (b) and the blur effect with HEVC (c). Our method (d) based on exact edges and homogeneous diffusion prevent this effect (contrast has been increased on depth maps for distortion visibility). Lower row: zoom on corresponding synthesized view V_4 without (e) or with JPEG-2000 (f) and HEVC (g) compressions and our method (h).

location respecting directional priorities.The border and seed values are also re-positioned following a predefined location.

B. Reconstructing the missing values by diffusion

We now have a sparse depth map containing only the edge, border and seed pixel values. A homogeneous diffusion-based inpainting approach is used to interpolate the missing data. This method is the simplest of the partial differential equations (PDEs) diffusion method, and has the advantage of low computational complexity. It directly stems from the heat equation:

$$\begin{cases} I_{t=0} = \widetilde{I} \\ \frac{\delta I}{\delta t} = \Delta I \end{cases}$$

where $\widetilde{I}$ is the decoded edge image before diffusion that will constitute the Dirichlet boundaries of the equation. The diffused data then satisfies the Laplace equation $\Delta I = 0$. The diffusion process is run in a hierarchical manner, each diffusion step being in addition helped with seeds and appropriate initialization. These three improvements have been introduced in the classical diffusion approach to limit the number of iterations required to converge, hence to speed up the entire process:

Hierarchical diffusion: A Gaussian pyramid is built from $\widetilde{I}$. The diffusion process is first performed on a lower level of the pyramid and the diffused values are then propagated to a higher level (3 levels are used and shown good performance). The propagation of the blurred version of the diffused pixel values from an lower level to an upper one helps to initialize the diffusion in unknown areas.

Middle range initialization: On the highest level, instead of starting from unknown value of $\widetilde{I}$ set at 0, we propose to initialize unknown values to the half of the possible range: 128 for an 8 bit depth map. This facilitates and speeds up the process of diffusion by limiting the number of required iterations to converge.

Seeding: As explained in section II-C, some seeds are chosen from a regular pattern both to accelerate the diffusion process and to provide accurate initialized values in large unknown areas. Indeed, this definitely achieves a fast and accurate diffusion -with a gain of 10 dB- for a quasi-exact reconstruction of the depth map.

IV. Experiments

A. Conditions

The performances of the proposed compression method are evaluated on an original resolution depth map from a MVD sequence "Breakdancers" from [7]. The depth maps were estimated through a color segmentation algorithm. The choice of this sequence is motivated by the presence of sharp object edges on depth maps. Additional results can be found on our webpage[1]. Most other raw MVD sequences might be suitable once they would be "sharpened" i.e. post-processed with a bilateral filter, as it is often required in practice [5].

B. Depth map quality evaluation

The reconstruction quality of our PDE-based method is investigated and compared with the JPEG2000 and HEVC-HM-4.1 Intra compressed versions. First, to illustrate the difference of quality reconstruction on edges, the three methods are compared at equal *Peak-Signal-to-Noise-Ratio (PSNR)*,(45 dB, JPEG-2000 with a Quality factor Q=25, HEVC-Q=40).

[1]http://www.irisa.fr/temics/staff/gautier/diffusion/

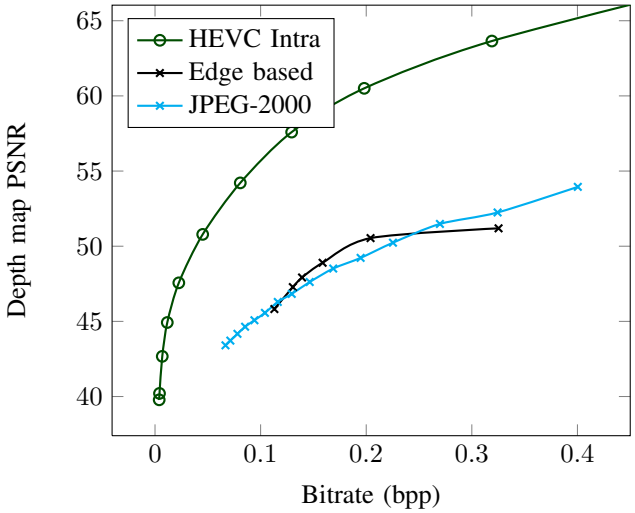

Fig. 3: Rate-Distortion performance of the V_3 depth map with different quality factors of JPEG-2000 and HEVC, and different Sobel detection thresholds λ of our method.

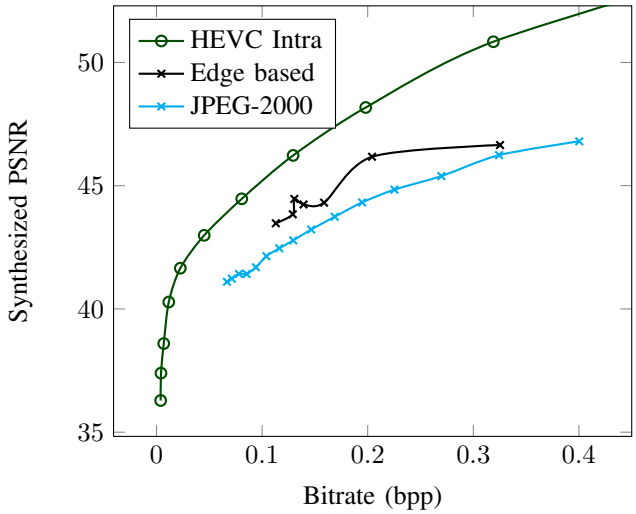

Fig. 4: Rate-Distortion performance of synthesized V_4 with the bitrate of V_3 depth map, for different quality factors of JPEG2000 and HEVC, and different Sobel detection thresholds of our method.

A zoom on the head of a character presenting initially sharp edges highlights the difference of edge quality depending on the compression type (Fig.2). While at high PSNR, the JPEG-2000 (a) and HEVC (b) versions of the depth map tend to blur the edges. This is commonly referred to as ringing artifacts. It appears with JPEG-2000 because of the lossy quantization following wavelet transformation. It might appears with HEVC because of deblocking filter limitation. Then both JPEG-2000 and HEVC cannot efficiently reconstitute the smooth gradient on uniform areas while preserving the edges. At the opposite, our proposed approach stores the exact edges and diffuses regions between these edges, resulting in a smooth gradient restitution on slanted surfaces and non distorted edges.

Thus we evaluate the global depth-map rate-distortion performances of the three encoding methods. Fig.3 shows that our approach outperforms JPEG-2000 except in very low or high bitrate conditions, while being under HEVC. No dedicated adjustment was perform in our method, only the threshold λ was varying to adjust its bitrate. An optimisation of the seed locations depending on bitrate might however improve the performance.

C. View synthesis quality evaluation

The impact of depth compression methods on rendering is measured by calculating the PSNR of a synthesized view (from a pair of uncompressed textures and compressed depth maps), with respect to an original synthesized view (from a pair of uncompressed textures and depth maps). The corresponding synthesized view from two original depth maps is then the reference. VSRS 3.0 [6] is used for view interpolation from this 2-view dataset. The R-D synthesis performance, illustrated in Fig.4, justifies the edge-coding approach over wavelet based encoders: undistorted edges permits an accurate and efficient view coding and rendering. Again, the PSNR measure shows its limitation of objective evaluation on perceived quality. Our method do not always outperform in term of rate-distorsion the existing methods (Fig.3, 4), but can improve the perceived quality of the synthesized view especially on edges.

V. Conclusion

We proposed a new method for lossless-edge depth map coding based on optimized path and fast homogeneous diffusion. Our method, combined with a Sobel edge detection, provides a simple but efficient compression of edges enabling a perfect restoration of the depth map contours. Then it outperforms JPEG-2000 in term of PSNR, while being competitive to HEVC in term of perceived quality. Thanks to careful edge selection and seeding, we also manage to increase the quality reconstruction of previous works based on edge image coding. Thus this lossless edge coding method could be locally applied to color image compression, especially on uniform areas. In this case the edge detection method should probably be optimized depending on edge smoothness. Finally, a depth map video encoder is in our scope for future research.

Acknowledgment

This work is supported by the ANR-PERSEE project.

References

[1] Z. Belhachmi, D. Bucur, B. Burgeth and J. Weickert, How to choose interpolation data in images, SIAM Journal on Applied Mathematics, 70, 1, 2009, 333-352

[2] Joint Bi-level Image Experts Group: Information technology - progressive lossy-lossless coding of bi-level images. ISO/IEC JTC1 11544, ITU-T Rec. T.82 (1993) Final Committee Draft 11544.

[3] M. Mainberger and J. Weickert, Edge-based image compression with homogeneous diffusion, Computer Analysis of Images and Patterns, 2009, 476-483

[4] P. Merkle, Y. Morvan,A. Smolic, D. Farin, K. Müller and T. Wiegand, The effects of multiview depth video compression on multiview rendering. Signal Processing: Image Communication, 24,1-2, 2009, pp73–88

[5] S. Smirnov, A. Gotchev, S. Sen, G. Tech and H. Brust, 3D Video Processing Algorithms–Part I, Tech. Rep., Mobile3DTV, 2010

[6] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima and Y. Mori, Reference softwares for depth estimation and view synthesis, ISO/IEC JTC1/SC29/WG11, Archamps, France, Tech. Rep. M15377, Apr. 2008.

[7] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder and R. Szeliski, High-quality video view interpolation using a layered representation, ACM SIGGRAPH. ACM New York, USA, 2004, pp. 600-608.