JOINT DEPTH/TEXTURE BIT-ALLOCATION FOR MULTI-VIEW VIDEO COMPRESSION

Yannick Morvan[1], *Dirk Farin*[1] *and Peter H. N. de With*[1,2]

[1] Eindhoven University of Technology, P.O. Box 513
5600 MB Eindhoven, Netherlands
{y.morvan,d.s.farin}@tue.nl

[2] LogicaCMG, RTSE, P.O. Box 7089
5605 JB Eindhoven, Netherlands
P.H.N.de.With@tue.nl

ABSTRACT

Multi-View display technology allows the presentation of a 3D video by showing simultaneously several views of the same scene. One approach to render these multiple views is to synthesize novel views using a Depth Image Based Rendering (DIBR) algorithm. Consequently, for the efficient transmission of 3D video signals, the compression of texture and also the depth images is required. Since the ratio between the depth and texture bit-rate is still an open question, we propose in this paper a novel joint depth/texture bit-allocation algorithm for the compression of multi-view video. The described algorithm combines the depth and texture rate-distortion (R-D) curves to obtain a single R-D surface that allows the optimization of the joint bit-allocation problem in relation to the obtained rendering quality. We subsequently discuss a fast hierarchical optimization algorithm that exploits the smooth monotonic properties of the R-D surface. The hierarchical optimization algorithm employs an orthogonal search pattern so that the number of image-compression iterations for measuring quality is minimized. Experimental results show an estimated gain of $1\ dB$ compared to an *ad-hoc* selection of bit-rates. Besides this, our joint model can be readily integrated into an MVC H.264 coder because it yields the optimal compression setting with a limited computation effort.

***Index Terms*—** 3D video, depth/texture compression, multi-view video coding, joint bit-allocation.

1. INTRODUCTION

The MPEG Multi-view Video Coding (MVC) group pursues solutions for the coding of 3D video. To build a compact representation of a 3D video, an approach currently investigated by MPEG relies on a depth-image based representation of the 3D scene. Such a representation combines a reference texture image with a corresponding depth image that describes the visible surface of objects in the 3D scene. Using a depth-image based representation, the 3D rendering of novel views can be subsequently performed, using image warping algorithms. Therefore, employing such a depth-image based representation involves the compression of multiple texture views and also their associated depth images.

Previous work on the compression of such a data set (texture and corresponding depth images), has treated the problem of texture [1] and depth [2, 3] compression independently. Such an independent coding yields high compression ratios of texture and depth data. However, the influence of texture and depth compression on 3D rendering was not incorporated in these experiments, so that the rendering quality trade-off was not considered. Furthermore, recent literature [4, 5] confirms that the trade-off between texture and depth bit-rate is not understood.

To illustrate the problem of joint compression of texture and depth, let us consider the two following cases. First assume that the texture and depth images are compressed at very high and low quality, respectively. In this case, detailed texture is mapped onto a coarse approximation of object surfaces, which thus yields rendering artifacts. Alternatively, when texture and depth images are compressed at low and high quality, respectively, a high quality depth image is employed to warp a coarsely quantized texture image, which also yields low-quality rendering. These two simple but extreme cases illustrate that a clear dependence exists between the texture and depth quality setting. Therefore, the quantization setting for both the depth and texture images, should be carefully selected. For this reason, we address in this paper, the following problem:
given a maximum bit-rate budget to represent the 3D scene, how to optimally distribute the bit-rate amongst the texture and the depth image such that the 3D rendering distortion is minimized ?

To solve this problem, we propose a new compression algorithm with a bit-rate control that unifies the texture and depth Rate-Distortion (R-D) functions. The attractiveness of the algorithm is that both depth and texture data are simultaneously combined into a joint R-D surface model that enables to find the optimal bit-allocation between texture and depth. We discuss the performance of the joint-coding optimization algorithm using an H.264 encoder, where we found that our joint model can be readily integrated as a practical sub-system, because it directly yields the optimal compression setting with a limited computation effort.

The remainder of this paper is structured as follows. Section 2 formulates the framework of the joint bit-allocation of texture and depth. Section 3 describes a fast hierarchical optimization algorithm. Experimental results are provided in Section 4 and the paper concludes with Section 5.

2. JOINT DEPTH/TEXTURE BIT-ALLOCATION

In this section, we first present a joint bit-allocation analysis of depth and texture and afterwards we provide an experimental analysis of the R-D surface to enable a fast optimization for high-quality rendering.

2.1. Joint bit-allocation problem formulation

Let us consider the problem of jointly coding a texture and depth image at a maximum rate R_{max} with minimum rendering distortion D_{render}. The rate R_{max} and distortion D_{render} functions can be defined as follows. First, the maximum rate value R_{max} can be decomposed into the sum of the texture and depth coding rate. Because the texture and depth images can be coded with two different quantizer settings (denoted q_t and q_d, respectively) the texture and depth rate functions can be written as $R_t(q_t)$ and $R_d(q_d)$, respectively. The joint rate function can therefore be written as

$$R_{max}(q_t, q_d) = R_t(q_t) + R_d(q_d).$$

Second, the rendering distortion function D_{render} depends on the Depth Image Based Rendering (DIBR) algorithm. The DIBR algorithm relies on the quality of the compressed texture and depth images and therefore on the quantization parameters q_t and q_d. Consequently, we define a joint rendering distortion as $D_{render}(q_t, q_d)$.

The goal of the joint bit-allocation is to determine the optimal quantization parameters (q_t^{opt}, q_d^{opt}) for coding the depth and texture images such that the rendering distortion is minimized. The optimization problem can now be formulated as follows:

$$(q_t^{opt}, q_d^{opt}) = \underset{q_d, q_t \in Q}{\arg\min}\, D_{render}(q_t, q_d), \tag{1}$$

under the constraint that

$$R_t(q_t^{opt}) + R_d(q_d^{opt}) \leq R_{max}$$

where Q denotes the set of all possible quantizer settings. Without prior assumption, the solution to Equation (1) involves an exhaustive search over Q, in order to find the quantization setting with minimum distortion. However, a more efficient search can be performed by exploiting special properties of the R-D function. For example, assuming a smooth monotonic R-D surface, hierarchical optimization methods can be employed. Therefore, prior to investigating fast search algorithms, we provide a performance-point analysis of the R-D function to validate the smoothness of the surface.

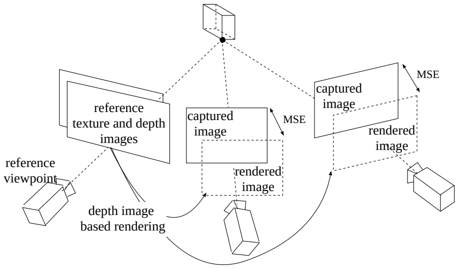
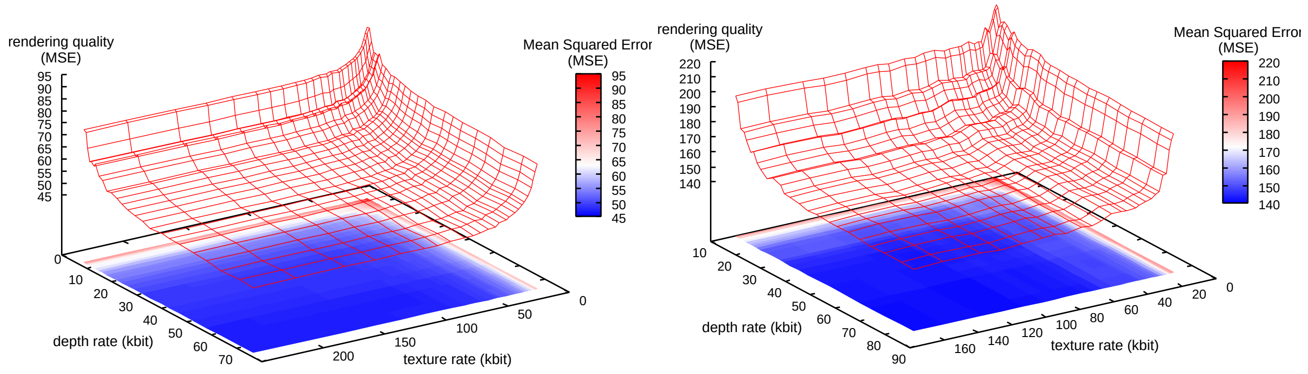


Fig. 1. The rendering distortion is obtained by rendering a synthetic image at the position of a neighboring camera. The rendering distortion is then evaluated by calculating the MSE between the original captured image and the rendered view.

2.2. R-D surface analysis

To analyse the R-D function, we construct a surface using an input data set composed of multi-view images and their corresponding depth images. The rendering algorithm is based on the relief-texture mapping [6]. It should be noted that the relief-texture rendering algorithm fills the disoccluded pixels by the background color, so that the rendered image does not show holes. Therefore, no special treatment is necessary to handle disoccluded pixels. We generate the R-D surface by measuring the rendering distortion for all quantizers (q_t, q_d) defined within a range search of $q_{min} \leq q_t, q_d \leq q_{max}$. In total, $k = q_{max} - q_{min} + 1$ compression iterations of the depth and texture images are carried out, which yields $k \times k$ R-D points. In our specific case, we employ an H.264 encoder to compress the reference texture and depth images. However, the proposed joint bit-allocation method is generic so that any depth and texture encoder can be employed.

To measure the rendering distortion, one solution is to warp a coded reference image using the corresponding depth image. The rendering distortion is evaluated by calculating the Mean Squared Error (MSE) between the rendered image and the corresponding image captured at the same location and orientation (see Figure 1). In MVC, the best selected quantizer setting has to be found for a data set with N views of the same scene. Therefore, considering an N-view dataset and a selected quantizer set (q_t, q_d), $N-1$ distortion measures can be obtained (excluding the reference image). To obtain a single rendering distortion measurement, the $N-1$ measures are then averaged. The pseudo-code of the R-D surface construction algorithm is summarized in Algorithm 1. As a result, Figure 2 shows the R-D surfaces for two images from the two MPEG multi-view sequences “Ballet” and “Breakdancers”. Considering Figure 2, it can be noted that both R-D surfaces show smooth monotonic properties. Up till now, we have not been able to define mathematical properties of the rendering function, so that we can only rely on the empirically found properties of the function. Assuming the previous holds, a fast bit-allocation can be employed.



(a)

(b)

Fig. 2. Figure 2(a) and Figure 2(b) depict the R-D surface for the sequence Breakdancers and Ballet, respectively. The color map corresponds to the height of the surface, i.e. the rendering quality.

Algorithm 1 R-D surface construction algorithm

```
Require: One reference texture and depth images.
Require: Neighboring views $V_i$ for distortion evaluation.
  initialize a 2D array RDSurface[][].
  for ($q_t = q_{min}; q_t <= q_{max}; q_t++$) do
    Encode the reference texture image at $QP = q_t$.
    for ($q_d = q_{min}; q_d <= q_{max}; q_d++$) do
      Encode the reference depth image at $QP = q_d$.
      for each non-reference view $V_i$ do
        Render an image at the position and orientation of the view $V_i$.
        Calculate MSE $m_i$ between captured and rendered image.
      end for
      $\bar{m}$=Average MSE $m_i$; RDSurface[$q_t$][$q_d$]=$\bar{m}$
    end for
  end for
```

3. HIERARCHICAL SEARCH OPTIMIZATION

The guiding principle of the hierarchical optimization is to perform a recursive, coarse-to-fine, search of a good quantizer setting. The algorithm can be summarized as follows. First, a search over a limited number of quantizer candidates (q_t, q_d) is performed. Practically, we employ nine candidates shown as black R-D points and organized them in a search pattern, as illustrated by Figure 3. Second, the algorithm selects the candidate with the lowest rendering distortion that satisfies the maximum bit-rate constraint. The search range is then refined and the process is recursively performed by using the selected quantizer set as an initialization, similar to the well-known *three-step search* in motion estimation. The minimum corresponds to the lowest distortion point after the last recursion.

This technique has two advantages. First, the hierarchical set-up of the search significantly reduces the computational complexity by reducing the set of quantizers candidates. Second, by employing an appropriate search pattern, the number of texture and depth images compression iterations can be decreased. For example, it can be observed in Figure 3 that the 3×3 orthogonal pattern of the black R-D points enable the re-use of depth and texture images so that only six compression iterations are required. Following this in the second step, by using the pre-defined grid, only four compressions of depth and texture images are necessary to obtain again nine R-D points (shown as dark grey in Figure 3). In contrast to this, a less-structured search such as a descent method would require a much larger number of image-compression operations. The hierarchical search algorithm is summarized in Algorithm 2.

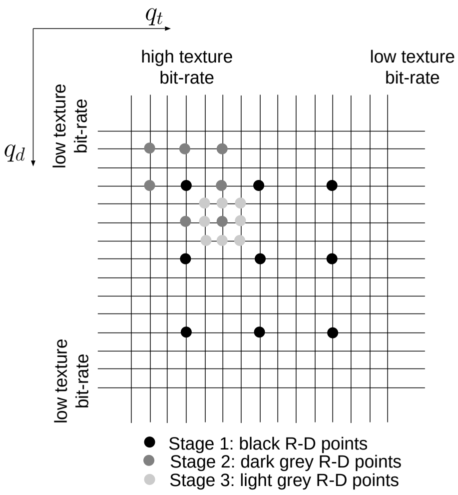
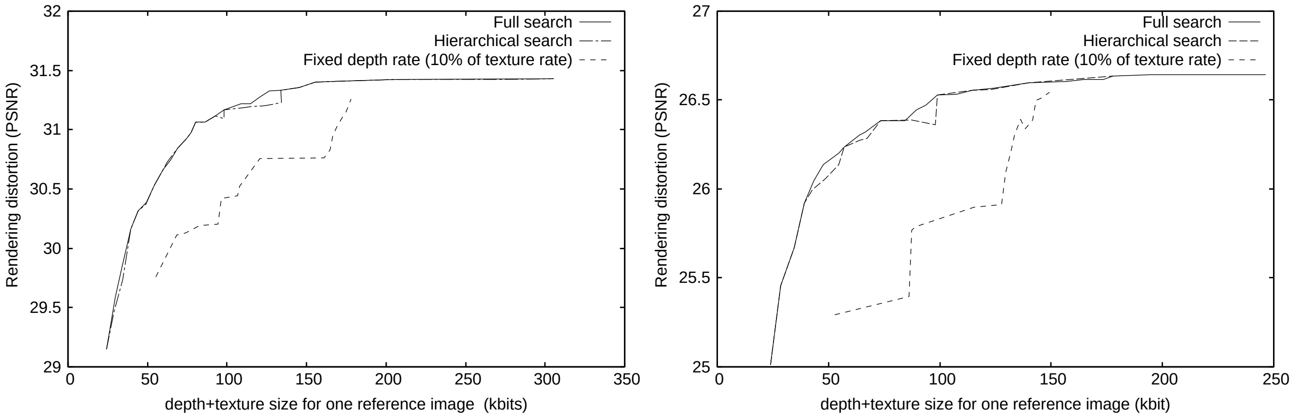


Fig. 3. Hierarchical search pattern of the appropriate quantization setting.

4. EXPERIMENTAL RESULTS

To evaluate the performance of the bit-allocation algorithm, experiments were carried out using a single multi-view image from the “Ballet” and “Breakdancers” sequences (1024×768). One reference depth and texture image of one view is compressed and the synthesized views are compared to the texture of the remaining views. To generate the R-D surfaces



(a) (b)

Fig. 4. Figure 4(a) and Figure 4(b) shows rendering quality for the sequence Breakdancers and Ballet, respectively. The R-D curve denoted "Fixed depth rate" is obtained by performing the rendering using a depth encoded at 10% of the texture bit-rate.

Algorithm 2 Hierarchical search - algorithm summary

```
Step 1: compress the depth and texture images and render views to generate the R-D points of the search pattern shown by Figure 3.
Step 2: select the R-D point that yields the lowest distortion and satisfies the constraint $R_t(q_t^{opt}) + R_d(q_d^{opt}) \leq R_{max}$.
Step 3: Initialize a new finer search pattern with halved step size around the previously selected R-D point.
Step 4: Go to Step 1 if the finest step size is not reached.
```

for both images, we set $q_t^{min} = q_d^{min} = 27$ and $q_t^{max} = q_d^{max} = 51$, so that 25×25 R-D points are obtained. For coding experiments, we employed the open-source H.264 encoder x264 [7]. The presented experiments attempt to quantify the rendering quality obtained using (a) a pre-defined depth bit-rate (corresponding to 10% of the texture bit-rate) or using a depth quantizer q_d determined by performing (b) a full search or (c) a hierarchical search. First, considering Figure 4, it can be observed that the proposed joint-bit allocation framework consistently out-performs the pre-defined depth bit-rate coding scheme. For example, observing Figure 4(a) and Figure 4(b), it can be seen that the joint bit-allocation framework yields a quality improvement of $0.8\ dB$ and $1\ dB$ at $75\ kbit$, respectively. Additionaly, employing the sub-optimal search does not sacrifice the rendering performance compared to full search. Thus, the sub-optimal hierarchical search provides a fast and accurate estimation of the optimal R-D point of operation.

5. CONCLUSIONS

In this paper, we have presented a joint depth/texture bit-allocation algorithm for the compression of multi-view images. To perform a joint bit-allocation optimization, we have proposed to combine both the depth and texture R-D curves into a single unified R-D surface. We have empirically verified that the R-D surface presents smooth monotonic properties so that fast optimization algorithms can be employed. A hierarchical search optimization of quantization parameters was implemented and experimental results reveal that the performance is comparable to a full-search parameter optimization. Because the algorithm features low computational complexity, the described joint bit-allocation optimization technique can be readily integrated into the MVC H.264 encoder currently developed within MPEG.

6. REFERENCES

[1] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image based rendering using 3-D scene geometry," *IEEE Trans. on Circ. and Syst. for Video Techno.*, pp. 1092–1106, Nov. 2003.

[2] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Stereo. Disp. and App. XVII, Proc. of the SPIE*, 2006.

[3] Y. Morvan, D. Farin, and P. H. N. de With, "Incorporating depth-image based view-prediction into H.264 for multiview-image coding," in *IEEE Int. Conf. on Image Proc. (ICIP) 2007*, San Antonio, to appear, 2007.

[4] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Image Commun.*, vol. 22, no. 2, pp. 217–234, 2007.

[5] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of h.264/avc for multiview video compression," in *IEEE Int. Conf. on Image Proc.*, Atlanta, 2006.

[6] M. M. Oliveira, "Relief-Texture Mapping", *Ph.D. diss.* UNC Computer Science, March 2000.

[7] "Webpage title: x264 a free H264/AVC encoder," http://developers.videolan.org/x264.html, last visited: June 2007.