Multiview Depth-Image Compression Using an Extended H.264 Encoder

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

[1] Eindhoven University of Technology, PO Box 513, 5600 MB, The Netherlands
[2] LogicaCMG, TSE, PO Box 7089, 5600 JB Eindhoven, The Netherlands

Abstract. This paper presents a predictive-coding algorithm for the compression of multiple depth-sequences obtained from a multi-camera acquisition setup. The proposed depth-prediction algorithm works by synthesizing a virtual depth-image that matches the depth-image (of the predicted camera). To generate this virtual depth-image, we use an image-rendering algorithm known as 3D image-warping. This newly proposed prediction technique is employed in a 3D coding system in order to compress multiview depth-sequences. For this purpose, we introduce an extended H.264 encoder that employs two prediction techniques: a block-based motion prediction and the previously mentioned 3D image-warping prediction. This extended H.264 encoder adaptively selects the most efficient prediction scheme for each image-block using a rate-distortion criterion. We present experimental results for several multiview depth-sequences, which show a quality improvement of about 2.5 dB as compared to H.264 inter-coded depth-images.

1 Introduction

The emerging 3D video technology enables novel applications such as 3D-TV or free-viewpoint video. A free-viewpoint video application provides the ability for users to interactively select a position (viewpoint) for viewing the scene. To render user-selected views of the video scene, various image-synthesis techniques have been developed [1]. The two major techniques use either a geometric model of the scene, or an interpolative model based on the neighboring cameras to generate a new user-selected view.

Recently, it has been shown that using a mixture of both techniques enables real-time free-viewpoint video rendering. One example of this [2] allows the synthesis of intermediate views along a chain of cameras. The algorithm estimates the epipolar geometry between each pair of successive cameras and rectifies the images pairwise. Disparity images are estimated for each pair of cameras and synthetic views are interpolated using an algorithm similar to the *View Morphing* [3] technique. A second example [4] employs a similar video capturing system composed of a set of multiple cameras. As opposed to the previous approach, the cameras are fully calibrated prior to the capture session (see Figure 1). Since the cameras are calibrated, the depth can be subsequently estimated for each view. Using the estimated depth-information, 3D warping techniques can be employed

J. Blanc-Talon et al. (Eds.): ACIVS 2007, LNCS 4678, pp. 675–686, 2007.

to perform view synthesis at the user-selected viewpoint. This selected virtual camera position is used to warp the two nearest-neighboring views by employing their corresponding depth images. Both warped views are finally blended to generate the final rendered image.

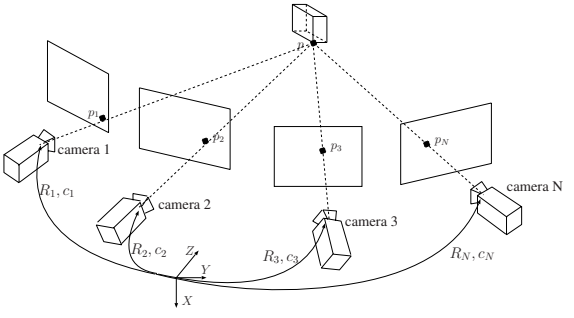

Fig. 1. Multiview capturing system in which the position and orientation of each camera is known. Because camera parameters are known, depth images can be estimated for each view and an image-warping algorithm can be used to synthesize virtual views.

Considering the transmission of 3D data, for both approaches, one depth-image for each view should be coded and transmitted. The major problem of this approach is that for each camera-view, an additional depth signal has to be transmitted. This leads to a considerable increase of the bitrate for transmitting 3D information. For example, an independent transmission of 8 depth-views of the "Breakdancers" sequence requires about 1.7 Mbit/s with a PSNR of 40 dB. This bitrate comes on top of the 10 Mbit/s for the multiview texture data. Therefore, a more efficient compression algorithm for transmitting depth-data is highly desirable, which is the key aspect of this paper.

Previous work on multiview depth-image compression has explored the idea that the estimated depth-images are highly correlated. As a result, a coding gain can be obtained by exploiting the inter-view dependency between the depth-sequences. To this end, two different approaches for predictive coding of depth-images have been investigated. A first depth-image prediction technique uses a block-based motion prediction [5]. The idea followed is to multiplex the depth-views such that a single video depth-stream is generated. The resulting video is then compressed using an H.264 encoder. A second, alternative depth-image prediction scheme [5] is based on an image-warping algorithm that synthesizes a depth-image as seen by the predicted camera. The advantage of a warping-based depth-image prediction is that the views can be accurately predicted, even when the baseline distance between the reference and predicted cameras is large, thereby yielding a high compression ratio.

In this paper, we propose a technique for coding multiple depth-sequences that employs predictive coding of depth-images. The depth-image prediction employs the two above-described algorithms, i.e. the block-based motion prediction and the image-warping prediction. The most efficient prediction method

is then selected for each image-block using a rate-distortion criterion. Because the prediction accuracy has a significant impact on the coding efficiency, we have implemented three different image-rendering algorithms for warping the depth-images:

1. simple 3D image warping, and
2. triangular-mesh-based rendering technique, and
3. *Relief Texture* [6] image warping.

Each of them has different rendering accuracy and computational complexity. First, the 3D image-warping technique performs image rendering at limited computing power by employing a simplified warping equation combined with several heuristic techniques. However, the quality of the rendered image is degraded, which thus results in a less accurate prediction. Second, the triangular-mesh-based technique aims at a high-quality rendered image by performing a sub-pixel warping algorithm. Reciprocally, such a precise algorithm is carried out at the cost of a high computational load. Third, an intermediate approach, i.e. relief-texture, decomposes the image-warping equation into a succession of simpler operations to obtain a computationally-efficient algorithm. For each image-rendering algorithm, we have conducted compression experiments and we present their coding gain. Experimental results show that the proposed depth-prediction algorithm yields up to 2.5 dB improvement when compared to H.264 inter-coded depth-images.

The remainder of this paper is organized as follows. Section 2 provides details about the warping-based depth-image prediction algorithms while Section 3 shows how the prediction algorithms can be integrated into an H.264 encoder. Experimental results are provided by Section 4 and the paper concludes with Section 5.

2 Warping-Based Depth-Image Prediction

In this section, we describe three alternative techniques for image warping that will be employed for depth-image prediction. First, we introduce the 3D image-warping [7] technique initially proposed by McMillan *et al.* and second, we describe a mesh-based image-rendering technique. Finally, a variant of the relief-texture mapping algorithm is proposed, that integrates the optics underlying real cameras.

2.1 Prediction Using 3D Image Warping

A single texture image and a corresponding depth-image are sufficient to synthesize novel views from arbitrary positions. Let us consider a 3D point at *homogeneous* world coordinates $\boldsymbol{P_w} = (X_w, Y_w, Z_w, 1)^T$ captured by two cameras and projected onto the reference and predicted image planes at pixel positions $\boldsymbol{p_1} = (x_1, y_1, 1)^T$ and $\boldsymbol{p_2} = (x_2, y_2, 1)^T$, respectively (see Figure 2). We assume that the reference camera is located at the coordinate-system origin and looks

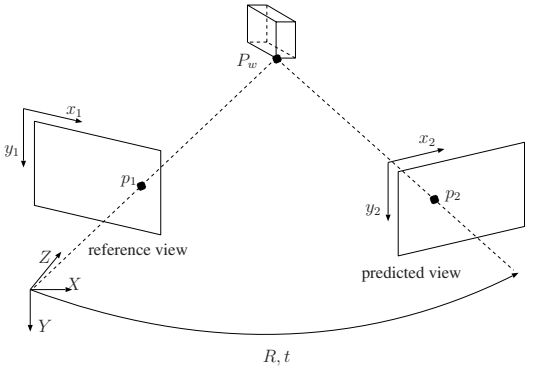

Fig. 2. Two projection points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ of a 3D point $\boldsymbol{P_w}$

along the Z-direction. The predicted camera location and orientation are described by its camera center $\boldsymbol{C_2}$ and the rotation matrix $\boldsymbol{R_2}$. This allows us to define the pixel positions $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ in both image planes by

$$\lambda_1 \boldsymbol{p_1} = [\boldsymbol{K_1}|\boldsymbol{0_3}]\, \boldsymbol{P_w} \ , \tag{1}$$

$$\lambda_2 \boldsymbol{p_2} = [\boldsymbol{K_2}|\boldsymbol{0_3}] \begin{bmatrix} \boldsymbol{R_2} & -\boldsymbol{R_2 C_2} \\ \boldsymbol{0_3^T} & 1 \end{bmatrix} \boldsymbol{P_w} = \boldsymbol{K_2 R_2} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} - \boldsymbol{K_2 R_2 C_2} \ , \tag{2}$$

where $\boldsymbol{K_1}$, $\boldsymbol{K_2}$ represent the 3×3 intrinsic parameter matrix of the corresponding cameras and λ_1, λ_2 some positive scaling factors [8]. Because the matrix K_1 is upper-triangular and $K_1(3,3) = 1$, the scaling factor λ_1 can be specified in this particular case by $\lambda_1 = Z_w$. From Equation (1), the 3D position of the original point $\boldsymbol{P_w}$ in the Euclidean domain can be written as

$$(X_w, Y_w, Z_w)^T = \boldsymbol{K_1^{-1}} \lambda_1 \boldsymbol{p_1} = \boldsymbol{K_1^{-1}} Z_w \boldsymbol{p_1}. \tag{3}$$

Finally, we obtain the predicted pixel position $\boldsymbol{p_2}$ by substituting Equation (3) into Equation (2) so that

$$\lambda_2 \boldsymbol{p_2} = \boldsymbol{K_2 R_2 K_1^{-1}} Z_w \boldsymbol{p_1} - \boldsymbol{K_2 R_2 C_2}. \tag{4}$$

Equation (4) constitutes the image-warping [7] equation that enables the synthesis of the predicted view from the original reference view and its corresponding depth-image. In the case that the world and reference-camera coordinate systems do not correspond, a coordinate-system conversion of the external camera parameters is performed. Similarly, the world depth-values Z_w are converted into the new reference coordinate system as well.

One issue of the previously described method is that input pixels $\boldsymbol{p_1}$ of the reference view are usually not mapped to a pixel $\boldsymbol{p_2}$ at integer pixel position. In our implementation, to obtain an integer pixel position, we simply map the sub-pixel coordinate $\boldsymbol{p_2}$ to the nearest integer pixel position $\hat{\boldsymbol{p_2}}$ with

$\hat{\boldsymbol{p}}_{\mathbf{2}} = (\hat{y_2}, \hat{x_2}, 1) = (\lfloor x_2 + 0.5 \rfloor, \lfloor y_2 + 0.5 \rfloor, 1)$. A second complication is that multiple original pixels can be projected onto the same pixel position in the predicted view. For example, a foreground pixel can occlude a background pixel in the interpolated view, which results in overlapping pixels. Additionally, some regions in the interpolated view are not visible from the original viewpoint, which results in holes in the predicted image. While the problem of overlapping pixels can be addressed using a technique called occlusion-compatible scanning order [7], undefined pixels in the predicted image cannot be analytically derived. Therefore, in our implementation, undefined pixels are padded using a simple pixel-copy of the nearest neighboring pixel. For simplicity, we defined a neighboring pixel as the nearest pixel in the image line. Although multiple heuristic techniques have been employed, experiments (see Section 4) have revealed that such a 3D image-warping generates depth-images with sufficient quality to perform predictive coding of depth-images.

2.2 Prediction Using Triangular Mesh

To avoid rendering artifacts such as occluded or undefined pixels, a natural approach to render 3D images is to employ a micro-triangular mesh. The idea is to triangulate the reference depth-image so that each triangle locally approximates the object surface. In our implementation, the depth-image triangulation is performed such that two micro-triangles per pixel are employed. For each triangle-vertex in the reference image, the corresponding position of the warped-vertex is calculated using Equation (4). Finally, a rasterization procedure is performed that converts the triangle-based geometric description of the warped image into a bitmap or raster image (see Figure 3). For efficient implementation, it can be noticed that each adjacent triangle shares two common vertices. Therefore, only one warped-vertex position per pixel needs to be computed to obtain the third warped-vertex position.

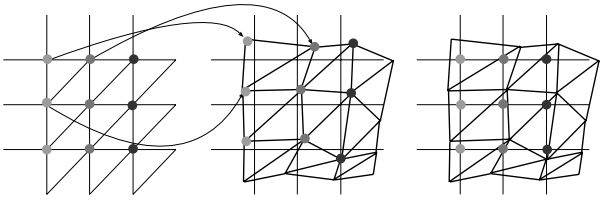

Fig. 3. Micro-triangular mesh rendering processing stages: first, each triangle vertex in the reference image is warped and, second, each triangle is rasterized to produce the output image

While such a technique leads to high-quality image-rendering, one disadvantage is the very large number of micro-triangles that involves a high computational complexity. As an alternative technique, relief-texture mapping has been introduced to reduce the polygonal count required in the warping procedure.

2.3 Prediction Using Relief Texture Mapping

The guiding principle of the relief-texture algorithm is to factorize the 3D image-warping equation into a combination of 2D texture-mapping operations. One well-known 2D texture-mapping operation corresponds to a perspective projection of planar texture onto a plane defined in a 3D world. Mathematically, this projection can be defined using homogeneous coordinates by a 3×3 matrix multiplication, and corresponds to a homography transform between two images. The advantage of using such a transformation is that a hardware implementation of this function is available in most Graphic Processor Units (GPU), so that processing time is dramatically reduced. Let us now factorize the warping function to obtain a homography transform in the factorization. From Equation (4), it can be derived that

$$\frac{\lambda_2}{Z_w}\boldsymbol{p_2} = \boldsymbol{K_2 R_2 K_1^{-1}} \cdot (\boldsymbol{p_1} - \frac{\boldsymbol{K_1 C_2}}{Z_w}). \tag{5}$$

Analyzing this equation, it can be seen that the first factor $\boldsymbol{K_2 R_2 K_1^{-1}}$ is equivalent to a 3×3 matrix and represents the desired homography transform.

Let us now analyze the second factor of the factorized equation, i.e. $(\boldsymbol{p_1} - \boldsymbol{K_1 C_2}/Z_w)$. This second factor projects the input pixel $\boldsymbol{p_1}$ onto an intermediate point $\boldsymbol{p_i} = (x_i, y_i, 1)^T$ that is defined by

$$\lambda_i \boldsymbol{p_i} = \boldsymbol{p_1} - \frac{\boldsymbol{K_1 C_2}}{Z_w}, \tag{6}$$

where λ_i defines a homogeneous scaling factor. It can be seen that this last operation performs the translation of the reference pixel $\boldsymbol{p_1}$ to the intermediate pixel $\boldsymbol{p_i}$. The translation vector can be expressed in homogeneous coordinates by

$$\lambda_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 - t_1 \\ y_1 - t_2 \\ 1 - t_3 \end{pmatrix} \text{ with } (t_1, t_2, t_3)^T = \frac{\boldsymbol{K_1 C_2}}{Z_w}. \tag{7}$$

Written in Euclidean coordinates, the intermediate pixel position is defined by

$$x_i = \frac{x_1 - t_1}{1 - t_3}, \qquad y_i = \frac{y_1 - t_2}{1 - t_3}. \tag{8}$$

It can be noticed that this result basically involves a 2D texture-mapping operation, which can be further decomposed into a sequence of two 1D transformations. In practice, these two 1D transformations are performed first, along rows, and second, along columns. This class of warping methods is known as scanline algorithms [9]. An advantage of this additional decomposition is that a simpler 1D texture-mapping algorithm can be employed (as opposed to 2D texture-mapping algorithms).

The synthesis of the view using relief-texture mapping is summarized as follows:

- Step 1: Perform warping of the reference depth-image along horizontal scanlines,
- Step 2: Perform warping of the (already horizontally-warped) depth-image along vertical scanlines,
- Step 3: Compute the planar projection of the intermediate depth-image using the homography transform defined by $\boldsymbol{K_2 R_2 K_1^{-1}}$ (for fast computing, exploit the GPU).

3 Incorporating Image Warping into an H.264 Encoder

We now propose a novel H.264 architecture dedicated to multiview coding that employs a block-based motion-prediction scheme and the previously explained image-warping prediction technique.

To integrate both warping-based prediction and block-based motion prediction, we have first added to the H.264 block-based motion-prediction algorithm a warping-based image prediction procedure, with the aim to select of one of the two according to some criterion. A disadvantage of such a multiview encoder is that the prediction error for the warping algorithm is not minimized, because high-quality warping does not necessarily lead to minimum prediction error. As a result, the compression efficiency is decreased. An alternative to selecting between two predictors, we employ a *combination* of the two predictors: (a) the warping-based predictor followed by (b) the block-based motion predictor (see Figure 4). The system concept becomes now as follows. First, we provide an approximation of the predicted view using image warping and, second, we refine the warping-based prediction using block-based motion prediction. In the refinement stage, the search for matching blocks is performed in a region of limited size, e.g. 16×16 pixels. For comparison, the motion disparity between two neighboring views in the "Ballet" sequence can be as high as 64×64 pixels. Figure 4 shows an overview of the described coding architecture.

Besides the compatibility with H.264 coding, the advantage of this approach is that the coding-mode selection can be performed for each image-block. More specifically, we employ three different coding modes in our multiview encoder. First, if the previously encoded depth-image D_{t-1} provides an accurate prediction of an image-block, D_{t-1} is selected as a reference. Alternatively, in the case the warped depth-image $W(D_{t-1})$ is sufficiently accurate, $W(D_{t-1})$ is selected as a a reference. Third, in the case the image-block cannot be accurately predicted using both previous prediction algorithms, the image-block is H.264 intra-coded as a fallback. This last case mostly occurs for occluded pixels that cannot be predicted with sufficient accuracy. To select the most appropriate coding mode, the same rate-distortion criterion that is employed in a standard H.264 encoder, is used. Thus, the H.264 standard offers suitable coding modes and an appropriate predictor-selection criterion to handle the various prediction accuracies of our algorithm.

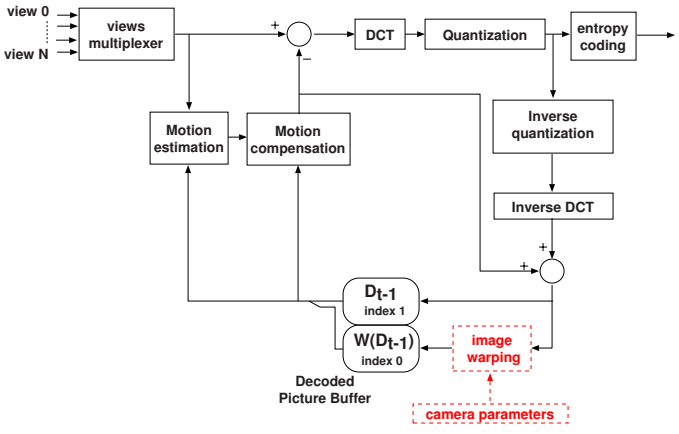

Fig. 4. Architecture of the extended H.264 encoder that adaptively employs the previously encoded depth image D_{t-1} or the corresponding warped image $W(D_{t-1})$ as reference frames

To enable the H.264 encoder using two different predictors, we employ two reference frames in the Decoded Picture Buffer (DPB) in the reconstruction loop: one reference for the warping-based prediction and a second for the block-based motion prediction (see Figure 4). However, the selection of the frame index in which each reference frame should be loaded in the DPB is important because of the following reason. In a standard H.264 encoder, the previously encoded frame (most correlated) is loaded in the DPB at index 0 and the "older" is available at index 1. This enables a "SKIP" coding mode that can be selected in the case the reference frame at index 0 in the DPB provides an accurate prediction. In this case, no quantized residual data or motion vectors are transmitted, thereby leading to a high coding efficiency. When using depth-images, our approach is to also load the most correlated depth-image in the reference frame buffers at index 0. Because the warping-based algorithm typically provides an accurate prediction, the warped depth-image should be loaded at index 0 while the previously encoded depth-image should be loaded at index 1 in the DPB. Consequently, a large number of image-blocks can be encoded using the "SKIP" coding mode (see Table 2).

Table 1 show a summary of possible coding modes employed in the extended H.264 encoder.

4 Experimental Results

For evaluating the performance of the coding algorithm, experiments were carried out using the "Ballet" and "Breakdancers" depth-sequences. The presented experiments investigate the impact of depth-prediction across multiple views. To measure the efficiency of the block-based motion-prediction algorithm, the

Table 1. Summary of possible coding modes and their corresponding description

Coding Mode	Description
Intra	Standard H.264 intra-coding
Inter-Block-Based-Motion	The previously encoded depth image D_{t-1} is selected as a reference. The image-block is H.264 inter-coded.
Inter-Warping	The warped depth-image $W(D_{t-1})$ is selected as a reference. The image-block is H.264 inter-coded.
Inter-Warping (SKIP mode)	The warped image provides a sufficiently accurate prediction such that the image-block is inter-coded, using the H.264 "SKIP" coding mode.

multiview depth-images were multiplexed and compressed by a standard H.264 encoder. To ensure that the temporal motion prediction does not interfere with the evaluation of the inter-view prediction algorithms, an intra-coded frame is inserted within each frame period. Figure 5 illustrates how the multiview depth-images are predicted using (1) block-based motion prediction only to obtain P-frames, or (2) an additional warping-based prediction to obtain P_w-frames.

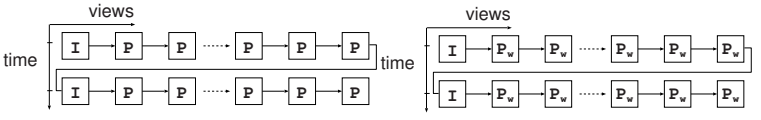

Fig. 5. (a) The multiple depth-images are predicted using a block-based motion prediction to obtain H.264 P-frames. (b) Depth-images are predicted using a block-based motion prediction and a warping-based image prediction to obtain P_w-frames.

Let us now discuss the obtained coding results using the extended H.264 coder and the above-given prediction structures. We perform the compression of depth-images under four different conditions. Depth-images are predicted using block-based motion estimation and subsequently one of the four options:

1. no additional warping-based prediction, i.e. original H.264 encoder ("Block-based prediction") or,
2. the 3D image-warping algorithm ("3D warping and block-based prediction") or,
3. the mesh-based rendering technique ("Triangular mesh and block-based prediction") or,
4. the relief-texture rendering algorithm ("relief texture and block-based prediction").

To measure the efficiency of the warping-based predictive-coding algorithms, we have implemented and inserted the three warping-based prediction algorithms

in the H.264 encoder. As described in Section 3, the warping-based prediction is followed by a prediction-error minimization. In our implementation, this refinement-minimization step is carried out by the H.264 block-based motion-compensation over a region of 16×16 pixels.

For coding experiments, we have employed the open-source H.264 encoder x264 [10]. The arithmetic coding algorithm CABAC was enabled for all experiments. For each sequence, the frame rate is 15 frames per second. Thus, the transmission of 8 views corresponds to a frame rate of 120 frames per second. Such a high frame rate explains the magnitude of the presented bitrates in Figure 6, ranging from approximatively 500 kbit/s to 5.5 Mbit/s.

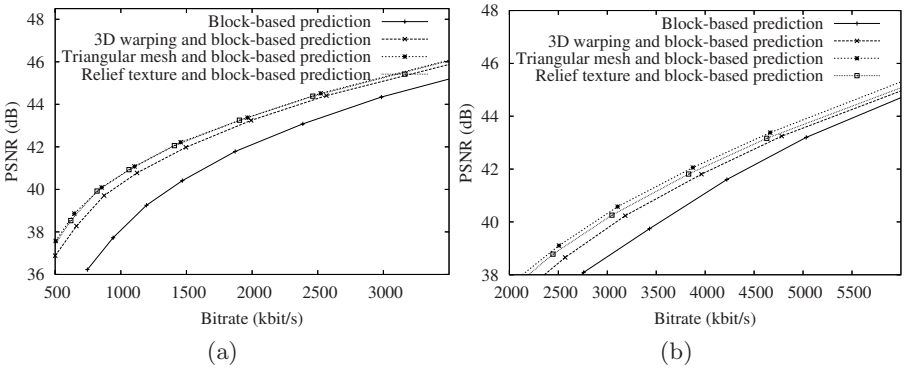

Fig. 6. Rate-distortion curves for encoding (a) the "Breakdancers" and (b) the "Ballet" depth-sequences

We produced the obtained rate-distortion curves of Figure 6(a) and Figure 6(b) under the parameters settings. First, it can be observed that all proposed warping-based prediction algorithms consistently outperform the standard block-based motion-prediction scheme. For example, considering Figure 6(a), it can be seen that the triangular-mesh rendering algorithm described in Section 2.2 yields a quality improvement of up to 2.5 dB over the block-based motion-prediction algorithm at 1 Mbits/s for the "Breakdancers" sequence. Additionally, although the "Ballet" multiview depth-sequence shows large occluded regions, a depth-image warping-based prediction yields a quality improvement of up to 1.5 dB at a bitrate of 3 Mbit/s. Let us now consider the two rate-distortion curves denoted "3D image warping and block-based motion" in Figure 6. Although multiple heuristic techniques have been employed to perform the 3D image-warping, a limited loss of quality of about 0.4 dB was observed at a bitrate of 1 Mbit/s and 3 Mbit/s for the sequences "Breakdancers" and "Ballet", respectively. For a low-complexity encoder, it is therefore appropriate to employ the image-warping technique from Section 2.1. Finally, while it has been discussed [6] that the relief-texture image-warping algorithm may produce rendering artifacts along depth-discontinuities, coding experiments show no significant coding

difference between a prediction performed using a triangular mesh or relief texture mapping. Therefore, relief texture can be effectively employed in a hardware implementation.

Observing Figure 7, it can be seen that occluded image-blocks at the right side of the two persons are intra-coded and sharp edges are encoded using a block-based motion prediction. Moreover, as can be noticed, the warping-based prediction provides a sufficiently accurate prediction in smooth areas. Because depth-images mainly consist of smooth regions, this coding mode is frequently selected. This observation is confirmed by the coding-mode selection statistics provided by Table 2.

Table 2. Coding-mode selection statistics using the triangular-mesh depth-image prediction

	Breakdancers	Ballet
Intra	8.1%	17.3 %
Inter-Block-Based-Motion	4.3%	5.4 %
Inter-Warping	3.3%	3.3%
Inter-Warping (SKIP mode)	84.3%	74.0%

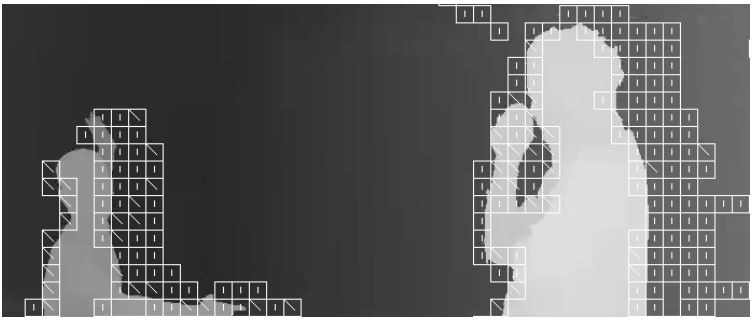

Fig. 7. Magnified area of one encoded depth-image from the "Ballet" sequence indicating the coding-mode selection. Coding modes "Intra" and "Inter-Block-Based-Motion" are refered to as a vertical line and a backward diagonal line, respectively. The coding mode "Inter-Warping" occupies the remaining space.

5 Conclusions

We have presented a new algorithm for the compression of multiview depth-images. The algorithm is based on extending the H.264 prediction by adding a rather accurate image-warping predictor. This approach leads to an extended H.264 encoder where the image warping is preceding the reference frame buffer in the reconstruction loop. Consequently, the depth-image is predicted using either a (1) a block-based motion prediction or (2) an image-warping predictor followed

by a block-based motion-prediction refinement. The selection of the prediction algorithm is optimized for each image-block using a rate-distortion criterion. Three image-warping techniques with different computational complexity have been integrated into an H.264 encoder and evaluated. Experimental results show that the most accurate image-warping algorithm leads to a quality improvement of up to 2.5 dB over the block-based motion-prediction algorithm. Additionally, it was found that the simplified 3D image-warping technique could synthesize a sufficiently accurate prediction of depth-images such that a quality improvement of 2.1 dB was obtained. Therefore, the presented technique demonstrates that an adaptive selection of different predictors can be beneficially employed to improve the compression of multiview depth-sequences, with a minor extension of the H.264 encoder.

References

1. Shum, H.Y., Kang, S.B.: Review of image-based rendering techniques. In: Proceedings of SPIE, Visual Communications and Image Processing, vol. 4067, pp. 2–13 (2000)
2. Farin, D., Morvan, Y., de With, P.H.N.: View interpolation along a chain of weakly calibrated cameras. In: IEEE Workshop on Content Generation and Coding for 3D-Television, IEEE Computer Society Press, Los Alamitos (2006)
3. Seitz, S.M., Dyer, C.R.: View morphing. In: SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 21–30. ACM Press, New York (1996)
4. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. ACM Transactions on Graphics 23(3), 600–608 (2004)
5. Morvan, Y., Farin, D., de With, P.H.N.: Prediction of depth images across multiple views. In: Proceedings of SPIE, Stereoscopic Displays and Applications (2007)
6. Oliveira, M.M.: Relief Texture Mapping. Ph.D. Dissertation. UNC Computer Science (March 2000)
7. McMillan, L.: An Image-Based Approach to Three-Dimensional Computer Graphics. University of North Carolina (April 1997)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
9. Wolberg, G.: Digital Image Warping. IEEE Computer Society Press, Los Alamitos (1990)
10. x264 a free H264/AVC encoder last visited: March (2007), `http://developers.videolan.org/x264.html`