SPATIALLY AND TEMPORALLY SCALABLE COMPRESSION OF ANIMATED 3D MESHES WITH MPEG-4 / FAMC

Nikolče Stefanoski, Jörn Ostermann

Institut für Informationsverarbeitung (TNT)
Leibniz Universität Hannover, Germany

ABSTRACT

We introduce an efficient method for scalable compression of animated 3D meshes. The approach consists of a combination of inter-frame motion compensation and layer-wise predictive coding of remaining residuals. It is shown that motion compensated predictive coding can lead to a significant improvement in compression performance upon the current state of the art with gains of over 40%. In addition, the created bit stream is temporally and spatially scalable allowing an adaptation to network transfer rates and end-user devices. This compression method is currently standardized within MPEG as part of MPEG-4 AFX Amd. 2, where it is referred to as FAMC - Frame-based Animated Mesh Compression.

Index Terms— Mesh compression, animation compression, dynamic mesh compression, MPEG-4, AFX.

1. INTRODUCTION

Animated 3D content is starting to become an integral part of many applications. It is already employed in numerous domains ranging from video games, CGI films, and special effects to scientific visualization and CAD. Applications in other prospective domains like 3D television and 3D cinema, immersive CAD, etc. already exist or are in development. In all these domains exists an increasing demand for efficient storage of animated 3D content. Furthermore, transmission of animated 3D content over different types of access networks (like the Internet, local area networks, or mobile networks) using different types of receiving devices (like PCs, laptops, PDAs, and smart phones) gains increasing importance. This imposes additional requirements to compressed data, since it has to be adaptable to network transfer rates and end-user devices.

FAMC (*Frame-based Animated Mesh Compression*), which is currently standardized within MPEG, accommodates to this requirements. It allows to efficiently compress sequences of static meshes of same connectivity in a scalable fashion by creating embedded scalable bit streams that allow layer-wise decoding and successive reconstruction of animated 3D content (Fig. 1).

In 1999 Lengyel presented the first method for compression of animated 3D meshes [1]. Since then several approaches were introduced, which can be classified into four groups: (1) deformation based approaches [1, 2, 3], (2) transform based approaches [4, 5], (3) predictive approaches [6, 7, 8, 9, 10], and (4) combinations of approaches of the first 3 groups [11, 12]. Only the predictive approaches [8, 9, 10] are capable for spatially and temporally scalable compression achieving efficient compression mainly at levels of high visual quality.

This work is partly supported by the EC within FP6 under Grant 511568 with the acronym 3DTV.

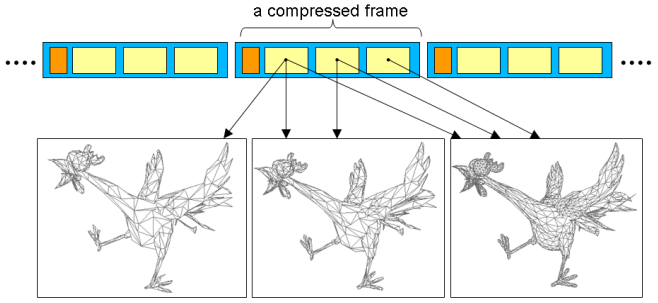

Fig. 1. Illustration of a part of an embedded spatially scalable bit-stream.

In this paper we will present the FAMC method for spatially and temporally scalable compression, which is based on [9, 10, 3]. It consists of a combination of a deformation based approach and a predictive approach. Compared to previous approaches the FAMC method allows increased compression efficiency at high as well as at lower levels of visual quality and provides a temporally and spatially scalable bit stream.

The rest of the paper is organized as follows. An overview of the proposed scalable FAMC method is given in Section 2 by describing in detail the components of the encoder. Compression results are evaluated and discussed in Section 3. Finally, we end with a conclusion in Section 4.

2. THE SCALABLE FAMC ENCODER

The proposed scalable FAMC encoder is illustrated in Fig. 2. The encoder has as input a sequence of static 3D meshes $\mathcal{F}_0, \ldots, \mathcal{F}_t, \ldots, \mathcal{F}_F$ with identical mesh connectivity, called frames. A frame $\mathcal{F}_t$ has always V vertices with 3D coordinates p_t^v assigned to each vertex v at instant t. Additional photometric attributes like vertex normals and vertex colors can be also encoded with the FAMC encoder. In the following we describe the encoding process only for vertex coordinates.

First, mesh connectivity and all 3D coordinates of $\mathcal{F}_0$ are encoded with a static mesh encoder. We employ here 3DMC [13], which is already part of the MPEG-4 standard. Subsequently, the first frame, frame $\mathcal{F}_0$, is exploited in the components *Motion-model designer* and *Layered decomposition designer* in order to extract information, which enables an efficient encoding of the remaining frames. Vertex coordinates of frames $\mathcal{F}_1, \ldots, \mathcal{F}_F$ provide input to a chain of three successive modules: (1) *Skinning-based motion compensation, (2) Layered prediction, and (3) CABAC*. In this processing chain inter- and intra-frame dependencies are exploited for achieving effi-

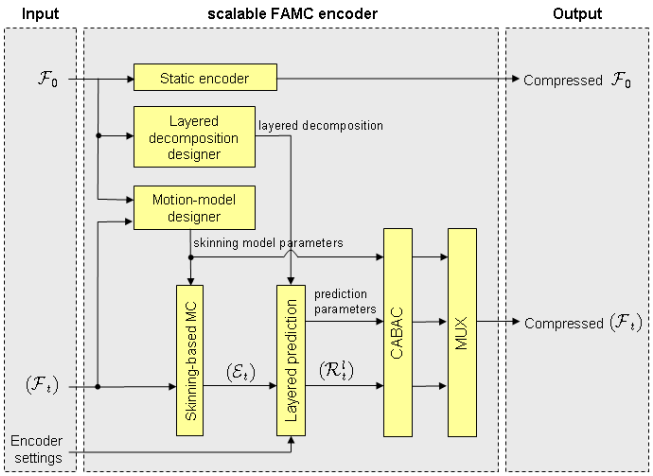

Fig. 2. A block diagram of the scalable FAMC encoder.

cient compression. First, motion is compensated from frame to frame using a skinning model. Subsequently, the residual signal is predictively encoded and the encoded data is organized in spatial and temporal layers. Finally, this data is entropy encoded and a multiplexer creates an embedded scalable bit-stream. In the following we detail the components of the scalable FAMC architecture.

2.1. Skinning-based motion compensation

First, skinning model parameters are calculated in the *Motion Model Designer* module. For this, mesh vertices are partitioned into a set of K clusters. The applied partitioning method guarantees that all 3D coordinates which are part of a cluster k in time instance t can be accurately described by a single 3D affine transform A_t^k relative to corresponding 3D coordinates in the first frame. Once the partition is determined, a skinning model is computed [3].

In the *Skinning-based motion compensation* module (Fig. 2) a predicted position $\hat{p}_t^v$ of a vertex v at instance t is specified, which is given by

$$\hat{p}_t^v = \left(\sum_{k=1}^{K} w_k^v \, A_t^k \right) p_0^v,$$

where w_k^v is a real-valued coefficient, so-called animation weight, which controls the influence of cluster k on the motion of the considered vertex v. The weight vector $\vec{w}^v = (w_1^v, \ldots, w_K^v)$, which leads to the smallest Euclidean error, is determined by finding the minimum of the following functional:

$$\Phi_v(\alpha_1, \ldots, \alpha_K) := \sum_{t=1}^{F} \left\| \left(\sum_{k=1}^{K} \alpha_k A_t^k \right) p_0^v - p_t^v \right\|^2 . \qquad (1)$$

The principle of linearly combining affine motions offers the advantage of obtaining a globally smooth motion field.

As a result of the skinning-based motion compensation step residuals

$$\varepsilon_t^v := p_t^v - \hat{p}_t^v \quad \forall\, t \in \{1, \ldots, F\},\ \forall\, v \in \{1, \ldots, V\}.$$

are obtained. In the end, sets of residuals $\mathcal{E}_1, \ldots, \mathcal{E}_F$ (residual frames) are provided as input to the next module and skinning model parameters $(\vec{w}^v)_{1 \le v \le V}$ and $(A_t^k)_{1 \le t \le F}^{1 \le k \le K}$ are encoded with CABAC.

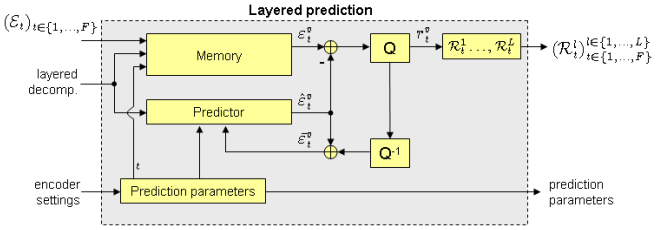

Fig. 3. A block diagram of the layered prediction module.

2.2. Layered prediction

Predictive coding is employed in the *Layered prediction* module to reduce remaining spatio-temporal dependencies between residuals. First, a so called *layered decomposition* is calculated in the *Layered decomposition designer* and *prediction parameters* are deduced from the *encoder settings*. The derivation of this data is described in Section 2.2.1. Subsequently, all residual frames $(\mathcal{E}_t)_{1 \le t \le F}$ are predictively encoded. Residual frames are not necessarily encoded in display order, i.e. $t = 1, \ldots, F$. In Section 2.2.3 frame encoding orders will be discussed.

Without loss of generality we assume now that residual frame $\mathcal{E}_t$ has to be encoded. All residuals $\varepsilon_t^{v_i}$ are predictively encoded in a predefined order $\varepsilon_t^{v_1} \ldots, \varepsilon_t^{v_V}$ using a DPCM loop (Fig. 3). Each predicted value $\hat{\varepsilon}_t^{v_i}$ is calculated based on already encoded residuals of the local spatio-temporal neighborhood. The residuals encoding order $\mathcal{O} = (v_1 \ldots, v_V)$ and the predicted residual value $\hat{\varepsilon}_t^{v_i}$ are determined with help of the *layered decomposition* and *prediction parameters*. Finally new residuals $r_t^{v_i}$ are determined, which are organized in L sets $R_t^1, \ldots, R_t^L$ and provided to the *CABAC* module for entropy encoding. These L sets per instant t define spatial layers and allow to create an embedded spatially scalable bit stream. Hence, spatial layers can be decoded successively at the decoder and allow for a gradual increase the spatial resolution per frame.

2.2.1. Layered decomposition and prediction parameters

A layered decomposition consists of pairs

$$\mathcal{LD}_i = (v_i, S_i) \quad \text{for} \quad 1 \le i \le V,$$

with $v_1, \ldots, v_V$ specifying the residuals encoding order $\mathcal{O}$, and S_i beeing a set of vertices in the neighborhood of vertex v_i with $S_i \subset \{v_1, \ldots, v_{i-1}\}$. This sequence of pairs $(\mathcal{LD}_i)_{1 \le i \le V}$ is calculated in the *Layered decomposition designer* using a mesh simplification algorithm, which is applied to the first frame [14, 10]. The order of vertex removal defines the reverse encoding order $\mathcal{O}' = (v_V, \ldots, v_i, \ldots, v_1)$, whereas each set S_i is defined as the set of neighboring vertices of vertex v_i before its removal. Thus, the information given with $(\mathcal{LD}_i)_{1 \le i \le V}$ allows to reverse the process of mesh simplification in the *Layered prediction* module, i.e. all residuals $\varepsilon_t^{v_i}$ are predictively encoded in order $\mathcal{O}$ whereas each $\hat{\varepsilon}_t^{v_i}$ is calculated based on already encoded residuals $\bar{\varepsilon}_t^u$ with $u \in S_{v_i}$.

Besides encoded residuals $\bar{\varepsilon}_t^u$ of instant t with $u \in S_{v_i}$ also corresponding residuals of other instances (reference frames) can be used for calculating a predicted value $\hat{\varepsilon}_t^{v_i}$ (Fig. 4). This additional information is specified by the prediction parameters. Prediction parameters indicate for each residual frame $\mathcal{E}_t$: the employed predictor type (linear or non-linear), the frame type (I, P, or B-frame), and associated instances of reference frames used for prediction. An I-frame has no reference frames, while a P- and a B-frame have respectively one and two reference frames.

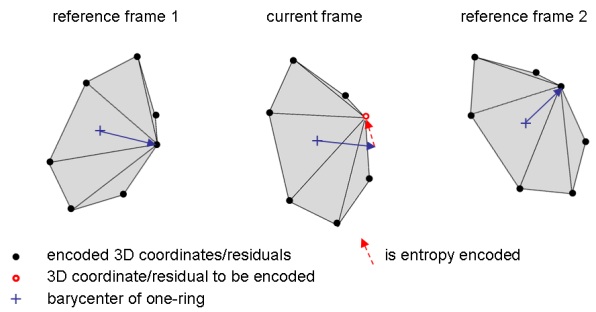

Fig. 4. Illustration of a B-frame predictor.

2.2.2. *Predictors*

Prediction parameters fix for each instant t: a prediction type, a frame type, and corresponding reference frames. All residuals $\varepsilon_t^{v_i}$ of instant t are then encoded in residuals encoding order $\mathcal{O}$ and for each residual a predicted value $\hat{\varepsilon}_t^v$ is calculated involving encoded residuals specified with help of S_v and reference frames. For instance, in a B-frame with reference frames r_1 and r_2 the following encoded residuals are exploited for prediction of ε_t^v: all $\bar{\varepsilon}_t^u$ with $u \in S_v$ and all $\bar{\varepsilon}_\tau^u$ with $\tau \in \{r_1, r_2\}$ and $u \in S_v \cup \{v\}$. In Figure 4 a linear B-frame predictor is illustrated. It predicts a residual ε_t^v by determining a correction vector, which is relative to the barycenter of neighboring residuals (indicated by S_v) in the current frame. The correction vector is calculated as the average of correction vectors determined in the two reference frames. For linear P-frame prediction a correction vector is calculated similar by using only one reference frame, while for I-frame prediction the correction vector is assumed to be a zero vector. Non-linear prediction is performed by representing correction vectors in local coordinate frames [10].

2.2.3. *Frame Encoding Order*

The MPEG-4/FAMC standard is designed to support arbitrary encoding orders for residual frames $(\mathcal{E}_t)_{t\in\{1,\ldots,F\}}$. This allows to encode residual frames also in an order which creates a temporally (and spatially) scalable bit stream. For this a hierarchical B-frame order is employed, which is illustrated in Figure 5 from top to bottom.

First, residual frame $\mathcal{E}_1$ is encoded as an I-frame. Thereafter residual frame $\mathcal{E}_{1+2^N}$ is encoded as a P-frame, with $N \in \mathbb{N}_0$ being a constant defined in the encoder settings. Thereafter, all frames in-between are encoded in the order depicted in Fig. 5. Following frames are encoded similar in groups of meshes (GOM) of size 2^N by employing the same frame structure. This GOM-wise frame encoding order provides a temporally scalable bit stream, since by skipping the last encoded frames of a GOM during decoding, a GOM is obtained with reduced frame rate.

Finally, all new residual frames $(R_t^1, \ldots, R_t^L)_t$, which are already organized in spatial layers, are provided to the CABAC module, with $t \in \{1, \ldots, F\}$ being in hierarchical B-frame order.

2.3. CABAC

In order to ensure an efficient entropy coding while keeping a low computational cost of the encoding/decoding processes, the CABAC (*Context-based Adaptive Binary Coding*) approach has been adopted in the MPEG-4 / FAMC standard [15]. It is retained from the MPEG-4 / AVC - H.264 standard [16] and it is well known for its mechanism for fast adaptation to statistical distributions.

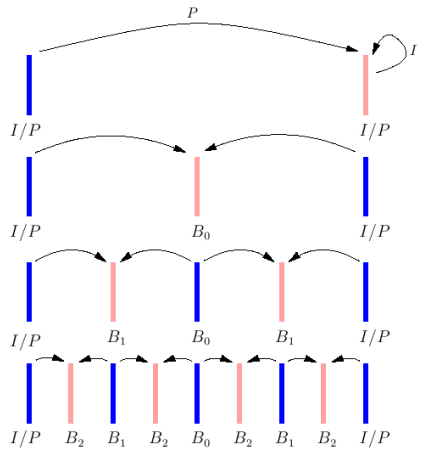

Fig. 5. Illustration of a hierarchical B-frame order with $N = 3$.

CABAC encodes all $(R_t^l)_{1\le t\le F}^{1\le l\le L}$ independently of each other and provides the encoded data to a multiplexer (Fig. 2), where a spatially and temporally scalable bit stream is created.

3. EVALUATION AND RESULTS

We evaluated the FAMC method using about 30 different mesh sequences [17] of various spatial resolution, length, and motion. In the following we show exemplary the evaluation results for mesh sequence CHICKEN consisting of 400 frames and 3030 vertices per frame (the CHICKEN character was created by Andrew Glassner et al., Microsoft Cooperation). The evaluation with the other sequences led to comparable results. We measure the bit rate in bits per vertex and frame (bpvf), while distortions between original and reconstructed mesh sequence are expressed using the KG error [4].

In Fig. 6 the influence of the level of spatial and temporal scalability to the bit rate is illustrated. Best compression efficiency is realized using 8 spatial layers combined with a GOM size greater or equal to 8. Thus, the support of spatial and temporal scalability leads to compression gains.

We evaluated the compression performance of the proposed scalable FAMC encoder using two variants which are supported in the standard: (1) FAMC (MC, LP), where motion compensation (MC) is performed as defined in Eq. 1, and (2) FAMC (no MC, LP), where no MC is performed, i.e. 3D coordinates are conducted directly to the layered prediction module (Fig. 2). A rate-distortion curve of a third FAMC variant, FAMC (MC, DCT) [15], which is also supported in the standard and does not support spatial and temporal scalability, is added in Fig. 7 for comparison reasons. We also compared the compression efficiency of the FAMC variants to the recently proposed compression methods TWC [5], MCDWT [11], and CPCA [12].

FAMC (MC, LP) exploits information from the whole mesh sequence to derive a skinning model and uses this data later for MC, while FAMC (no MC, LP) encodes frames in GOM-by-GOM fashion by exploiting only dependencies within a GOM and between consecutive GOMs, i.e. no global view to the whole sequence is needed. This leads to a fast encoding process and low memory requirements. On the other hand, the exploitation of a skinning model leads to additional gains in bit rate of 15% at an error of 0.044% (Fig. 7). FAMC (MC, DCT) shows best overall performance at the expense of spatial and temporal scalability. In the domain of very high quality (errors below 0.015%) all three FAMC variants show almost the same

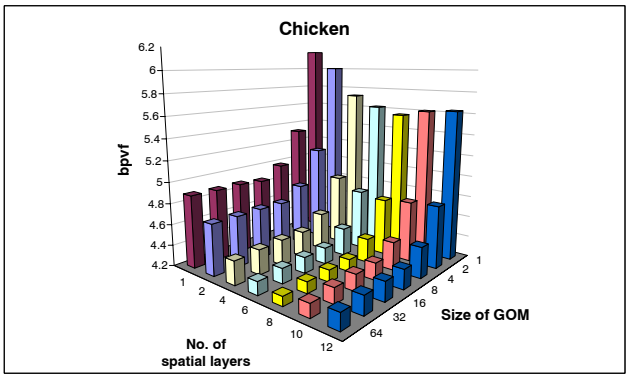

Fig. 6. Impact of the level of spatial and temporal scalability on the coding efficiency at a fixed KG-error of 0.044 %.

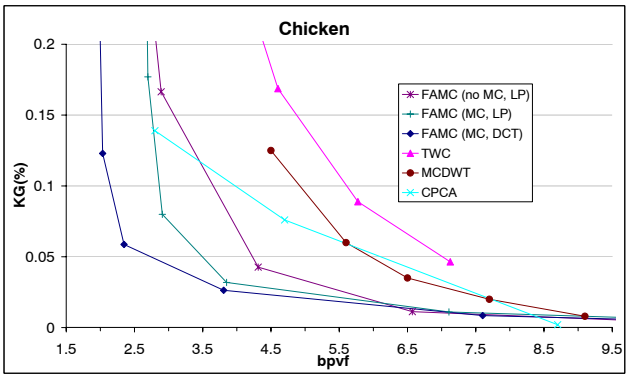

Fig. 7. Comparative compression results. A KG error of less than 0.044% can be regarded as lossless with regard to visual quality.

compression efficiency. TWC, MCTWC, and CPCA are significantly outperformed by FAMC in almost all error domains, e.g. FAMC (no MC, LP) and FAMC (MC, LP) achieve gains in bit-rate of over 30% and 40% respectively at an error of 0.044% when compared to this approaches.

4. CONCLUSION

We have introduced the scalable FAMC method for animated 3D meshes. In the evaluation we have shown that spatial and temporal scalability leads to increased compression performance. We have shown that layer-wise predictive coding leads to gains of 30% upon the current state of the art in domains of high visual quality and provides features like spatial and temporal scalability, fast encoding, and low memory requirement. Furthermore, we have shown that a global view to the mesh sequence can be exploited for efficient motion compensation leading to additional gains of 15%.

5. REFERENCES

[1] Jerome Edward Lengyel, "Compression of time-dependent geometry," in *Symposium on Interactive 3D graphics*, New York, NY, USA, 1999, pp. 89–95, ACM Press.

[2] K. Müller, A. Smolic, M. Kautzner, P. Eisert, and T. Wiegand, "Rate-distortion optimization in dynamic mesh compression," in *Proc. the IEEE International Conference on Image Processing*, Atlanta, USA, 2006, pp. 533–536.

[3] Khaled Mamou, Titus Zaharia, and Francoise Prêteux, "A skinning approach for dynamic 3D mesh compression," *Comput. Animat. Virtual Worlds*, vol. 17, no. 3–4, pp. 337–346, 2006.

[4] Z. Karni and C. Gotsman, "Compression of soft-body animation sequences," in *Computers & Graphics 28, 1*, 2004, pp. 25–34.

[5] Frédéric Payan and Marc Antonini, "Temporal wavelet-based compression for 3D animated models," *Computers & Graphics*, vol. 31, no. 1, pp. 77–88, 2007.

[6] Lawrence Ibarria and Jarek Rossignac, "Dynapack: space-time compression of the 3D animations of triangle meshes with fixed connectivity," in *Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, Switzerland, Switzerland, 2003, pp. 126–135.

[7] Nikolce Stefanoski and Jörn Ostermann, "Connectivity-guided predictive compression of dynamic 3D meshes," in *Proc. of the IEEE International Conference on Image Processing*, Oct 2006.

[8] I. Guskov and A. Khodakovsky, "Wavelet compression of parametrically coherent mesh sequences," in *Eurographics Symposium on Computer Animation*, Aug 2004, pp. 183–192.

[9] Nikolce Stefanoski, Xiaoliang Liu, Patrick Klie, and Jörn Ostermann, "Scalable linear predictive coding of time-consistent 3D mesh sequences," in *Proc. of 3DTV-CON, The True Vision - Capture, Transmission and Display of 3D Video*, May 2007.

[10] Nikolce Stefanoski, Patrick Klie, Xiaoliang Liu, and Jörn Ostermann, "Layered coding of time-consistent dynamic 3D meshes using a non-linear predictor," in *Proc. of the IEEE International Conference on Image Processing*, Sep 2007.

[11] Y. Boulfani-Cuisinaud and M. Antonini, "Motion-based geometry compensation for dwt compression of 3D mesh sequence," in *IEEE International Conference in Image Processing (CD-ROM)*, Texas, USA, 2007.

[12] Mirko Sattler, Ralf Sarlette, and Reinhard Klein, "Simple and efficient compression of animation sequences," in *Proc. of the ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2005, pp. 209–217, ACM Press.

[13] ISO/IEC JTC1/SC29/WG11, "Information technology - coding of audio-visual objects. part 2: Visual.," MPEG, Doc. N4350, Sydney, Australia, 2001.

[14] N. Stefanoski and J. Ostermann, "Scalable compression of dynamic 3D meshes," MPEG, Doc. M14363, San Jose, USA, April 2007.

[15] Khaled Mamamou, Titus Zaharia, and Françoise Prêteux, "FAMC: The MPEG-4 standard for animated mesh compression," in *submitted to Proc. of the IEEE International Conference on Image Processing*, Oct 2008.

[16] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in h.264/avc video compression standard," *IEEE Transactions on Circuits Systems for Video Technolgy*, vol. 13, no. 7, pp. 620–636, 2003.

[17] Marius Preda, "3D graphics compression core experiments description," *ISO/IEC JTC 1/SC 29/WG 11 N8499*, 2006.