THE NEW MPEG-4/FAMC STANDARD FOR ANIMATED 3D MESH COMPRESSION

K. Mamou[1]*, N. Stefanoski*[2]*, H. Kirchhoffer*[3] *K. Müller*[3]*, T. Zaharia*[1]*, F. Preteux*[1]*, D. Marpe*[3]*, J. Ostermann*[2]

[1]ARTEMIS Department, Institut TELECOM / TELECOM & Management SudParis, Evry, France
[2]Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover, Hannover, Germany
[3]Fraunhofer Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, Berlin, Germany

ABSTRACT

This paper presents a new compression technique for 3D dynamic meshes, referred to as FAMC – Frame-based Animated Mesh Compression, recently promoted within the MPEG-4 standard as Amendement 2 of part 16 (AFX – *Animation Framework eXtension*). The FAMC approach combines a model-based motion-compensation strategy with transform/predictive coding of residual errors. First, a skinning motion-compensation model is automatically derived from a frame-based representation. Subsequently, either 1) DCT/lifting wavelets or 2) layer-based predictive coding is employed to exploit remaining spatio-temporal correlations in the residual signal. Both motion model parameters and residual signal components are finally encoded by using context-based adaptive binary arithmetic coding (CABAC). The proposed FAMC encoder offers high compression performance with gains of 60% in terms of bit-rate savings relative to previous MPEG-4 technology and of 20% to 40% relative to state-of-the-art techniques. FAMC is well suited for compressing both geometric and photometric (normal vectors, colors...) attributes. In addition, FAMC also supports a rich set of functionalities including streaming, scalability (spatial, temporal and quality) and progressive transmission.

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

1. INTRODUCTION

Animated 3D content is nowadays an integral part of numerous general public, entertainment, educational and professional applications with high socio-economic impact, related to the industries of video games, CGI films, special effects, and CAD systems.

Most often, 3D animations are represented as dynamic 3D meshes with constant connectivity and time-varying geometry, which are stored in a key-frame-based format (*i.e.* a static 3D mesh for each key-frame). Efficiently storing, transmitting and rendering such a memory consuming representation becomes a major challenge, as testifies the rich literature dedicated to this emerging research area (see [1] for an overview).

In addition, within the more general framework of convergence of fixed and mobile technologies, modern industrial applications should respond to the paradigms of universal access and content re-use. Content in general, and 3D content in particular should be available anytime and anywhere, whatever the user's terminals (PC, laptop, PDA, mobile phone), and communication networks involved. From a methodological point of view, such requirements translate into functionalities of scalable/progressive compression, for transmitting/broadcasting 3D animation sequences on different fixed/mobile communication channels with various bandwidths, and scalable rendering, for guaranteeing the effective visualization of 3D content on a large scale of terminals, including devices with low computing and memory capabilities.

The issue of compressing dynamic 3D meshes has been first considered by Lengyel in 1999 [2]. Since Lengyel pioneering work, numerous technical and methodological contributions have been proposed. They can be structured within the following four families: (1) Local spatio-temporal predictive approaches [3]; (2) Principal Component Analysis (PCA)-based techniques [4, 5, 6]; (3) Wavelet-based methods [7, 8]; and (4) Segmentation-based approaches [9, 10].

The FAMC (*Frame-based Animated Mesh Compression*) method proposed in this paper and recently adopted by the MPEG-4/AFX standard [11] exploits: (1) a skinning-based motion compensation model, automatically and optimally derived from arbitrary key-frame representations, (2) a temporal transform (DCT or wavelets) and/or layered-based prediction [12] for compressing the remaining spatio-temporal correlations in the residual signal, and (3) a CABAC arithmetic encoder [13] to ensure an efficient binary encoding at a low computational complexity.

The rest of the paper is organized as follows. The FAMC method, with encoding algorithms and coded representations is described in detail in Section 2. In Section 3 we discuss and analyze how the proposed FAMC encoding scheme responds to streaming, progressive transmission, and scalable rendering functionalities. The objective experimental evaluation carried out on the MPEG-4 test data set is presented in Section 4. Finally, Section 5 concludes the paper ans opens perspectives of future work.

2. OVERVIEW OF THE FAMC CODER

The proposed FAMC encoder architecture is illustrated in Figure 1. The encoder has as input a sequence of key frames (static 3D meshes) denoted by $\mathcal{F}_0, \ldots, \mathcal{F}_t, \ldots, \mathcal{F}_F$ with identical mesh connectivity. Let us denote χ_t^v the 3D coordinates of vertex v at frame t, and V the total number of vertices.

First, mesh connectivity and 3D coordinates of the first frame $\mathcal{F}_0$ are encoded with a static mesh encoder (*e.g.*, AFX-3DMC [14]). Subsequently the first frame is exploited in the *Motion-model designer* and *Layered decomposition designer* modules. Vertices coordinates and optionally photometric attributes such as normals and colors of frames $\mathcal{F}_0, \ldots, \mathcal{F}_F$ provide input to a chain of four successive modules: (1) *Skinning-based motion compensation, (2) Transform, (3) Layered prediction, and (4) CABAC*. Inter- and intra-frame dependencies are here exploited for achieving efficient compression. Let us now detail each component of the FAMC architecture, by starting with the skinning-based motion modeling and compensation

[2] This work is partly supported by the EC within FP6 under grant 511568 with the acronym 3DTV.

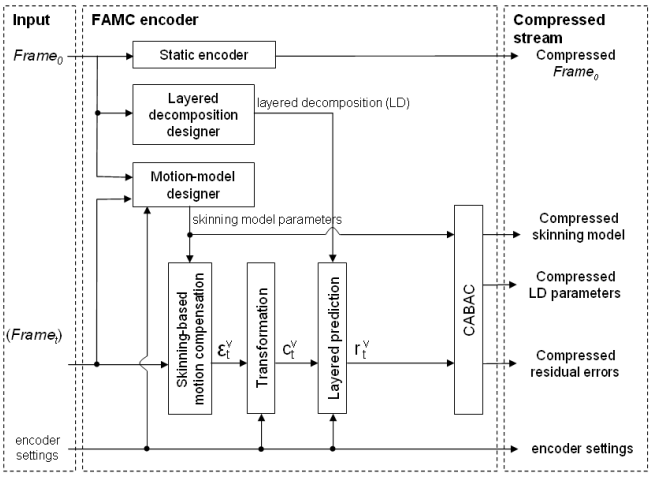

Fig. 1. Synopsis of the FAMC encoding algorithm.

modules.

2.1. Skinning-based motion compensation

The skinning motion model is determined in the the *Motion Model Designer* module. The mesh vertices are first optimally partitioned [15] into a set of K clusters such that the motion of each cluster can be accurately described by a single 3D affine transform A_t^k, associated to each cluster k and time instance t. Then, a skinning-based model prediction is defined as [16]:

$$\widehat{\chi}_t^v = \sum_{k=1}^{K} w_v^k \; A_t^k \; \chi_0^v,$$

where $\widehat{\chi}_t^v$ denotes the predicted position of a vertex v at frame t, and w_k^v is a real-valued coefficient, so-called animation weight, which controls the influence of the patch k on the motion of the considered vertex v. The optimal (in the L_2 sense) weight vector $w^v = (w_k^v)_{k\in\{1,\ldots,K\}}$ is expressed as:

$$w^v = \arg\min_{\alpha\in R^K} \sum_{t=1}^{F} \left\| \sum_{k=1}^{K} \alpha_k A_t^k \chi_0^v - \chi_t^v \right\|^2.$$

The *skinning-based motion compensation* module determines the prediction errors, defined as:

$$\forall\, t \in \{1,\ldots,F\},\ \forall\, v \in \{1,\ldots,V\},\ \varepsilon_t^v = \chi_t^v - \widehat{\chi}_t^v.$$

Normal vectors are often associated with mesh vertices within the framework of real-time and smooth rendering applications. The same motion model parameters determined for 3D coordinates can also be exploited for predicting normals. The following normal vector predictor is considered in this case:

$$\widehat{N}_t^v = (U_t^v \times W_t^v) / \left\| \sum_{k=1}^{K} U_t^v \times W_t^v \right\|,$$

where (U_1^v, W_1^v, N_1^v) represents an orthonormal basis of R^3, constructed by selecting two orthogonal vectors U_1^v and V_1^v both orthogonal to the normal vector N_1^v of the v vertex at the first frame of

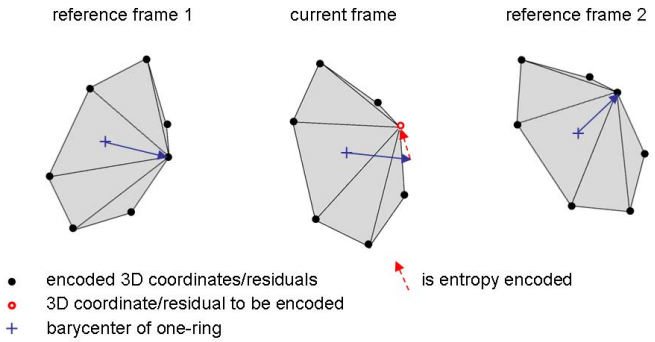

Fig. 2. Illustration of a B-frame predictor.

the sequence, with :

$$U_t^v = \frac{\sum_{k=1}^{K} w_v^k A_t^k U_1^v}{\left\| \sum_{k=1}^{K} w_v^k A_t^k U_1^v \right\|}, \text{ and } W_t^v = \frac{\sum_{k=1}^{K} w_v^k A_t^k W_1^v}{\left\| \sum_{k=1}^{K} w_v^k A_t^k W_1^v \right\|}.$$

The proposed predictor operates within the space of tangent vectors, instead of directly treating the normals, which makes it possible to overcome normalization to unity issues and ensures the predictor's optimality in the case of affine motions.

2.2. Transform

Residual temporal correlations within the prediction error signal ε_t^v are reduced by applying 1D transform in temporal direction. For each vertex v three 1D transforms (one for each x, y and z coordinate) are applied to the residual vector $(\varepsilon_1^v, \ldots, \varepsilon_F^v)$. Let $(c_t^v)_{t\in\{1,\ldots,F}$ denote the spectral coefficients associated to vertex v. FAMC supports the following three transforms: (1) DCT, (2) integer-to-integer (4-2) bi-orthogonal wavelet transform implemented through lifting scheme [17], and (3) bypass, i.e. $c_t^v = \varepsilon_t^v$. The emplyed transform is specifies by the *encoder settings*. The transformed residuals (c_t^v) feed the layered prediction module described in the next section.

2.3. Layered prediction

In this module, the remaining spatio-temporal dependencies between coefficients c_t^v are eliminated by achieving an additional local spatio-temporal prediction stage. A traversal order of the mesh vertices, denoted by $\mathcal{O} = \{i_1, ..., i_V\}$, is first determined for the first frame of the animation by applying a sequence of edge-collapse operations [12]. Coefficients at time instance t are then encoded in the reverse order $\mathcal{O}' = \{i_V, ..., i_1\}$ using a DPCM loop. Thereby, already encoded coefficients of the one-ring neighborhood of the current coefficient are exploited for its prediction (Figure 2).

In order to specify a predictor for a time instance t, three prediction modes can be selected, *i.e.* I-frame prediction, which exploits only encoded coefficients of the current frame for prediction (no reference frames), and P- and B-frame prediction, which additionally exploits encoded coefficients of one or two reference frames, respectively. In the case of B-frame predictors illustrated Figure 2, a coefficient is predicted by determining a correction vector, which is relative to the one-ring barycenter of the current frame. The correction vector is thereby calculated as the average of correction vectors determined in reference frames. A predicted coefficient $\hat{c}_t^v$ is then obtained. The selected prediction mode (I, P, or B) and used time

instances for reference frames are encoded for each frame as side information.

This component of the FAMC coder allows to reconstruct a frame at the decoder side by successively increasing its spatial resolution through a sequence of successive inverse simplification operations. Multiple spatial resolution layers are thus constructed.

Finally, prediction errors $r_t^v = c_t^v - \hat{c}_t^v$ are calculated, grouped into layers, uniformly quantized, and provided as input to the *CABAC* module.

2.4. CABAC

In the initially proposed version of FAMC [18], statistical coding of the individual FAMC information parts was performed by using an N-ary or *multialphabet* arithmetic coder with an *a priori* unknown maximum alphabet size N. Multialphabet arithmetic coding, however, is known to be costly, both in terms of computational and modeling costs, in particular in cases where the actual number of different symbols to encode may be considerably smaller than N.

Context-based Adaptive Binary Arithmetic Coding (CABAC), on the other hand, has proven to be an efficient technique of statistical coding in the area of video coding [13]. It handles multiple sources with different alphabet sizes and different statistical properties by application of a three-step process consisting in binarization, context modeling, and binary arithmetic coding. By using a computationally efficient, multiplication-free binary arithmetic coding engine along with a table-based probability estimator and by tuning the binarization and context modeling schemes to the individual characteristics of the given subsources, a high degree of coding efficiency can be achieved with rather moderate computational costs [13].

CABAC has been integrated into FAMC along the aforementioned basic principle and by employing the corresponding binary arithmetic coding engine, as specified in H.264/AVC. Appropriate binarization and context modeling schemes have been designed in order to be partly configurable by the encoder, thus providing a close match to the observed statistical properties of each component in FAMC. These components consist of: (1) the skinning model, consisting of the partition, a set of 3D affine transforms associated with the partition's clusters, and animation weights associated to each vertex of the mesh, (2) prediction parameters, which specify per frame the used prediction mode (I, P, or B) with reference frames as well as prediction type (linear or non-linear) and other predictor side information, and (3) residual errors in the transform domain, which are optionally grouped into layers before encoding. Layer-wise encoding may provide an embedded bit-stream supporting different types of scalability (*cf.* Section 3). More details of the CABAC-based approach in FAMC can be found in [19].

3. FUNCTIONALITIES

The FAMC encoder supports different functionalities, depending on the selected encoder settings for the *Transform* (DCT, Lift, or bypass) and the *Layered prediction* module (LD or bypass). Combinations of these encoder settings lead to bit streams adapted for different functionalities.

Both *Transform* and the *Layered prediction* modules are structuring the output signal into layers, which induces different types of scalability. Bit streams created with DCT or Lift setting provide quality scalability, since successive decoding of transform coefficients allows a reconstruction of the animation with increasing quality, without changing its spatial or temporal resolution. On the other hand, an encoder with LD setting creates a spatially scalable bit stream, since residuals of each frame are grouped into the spatial layers defined by the layered decomposition. Furthermore, the LD setting supports also temporal scalability, when frames are encoded in hierarchical B-frame order [20, 21]. This allows to decode a fraction of the bit stream, giving a reconstructed animation with reduced frame rate. Combined settings (DCT+LD and Lift+LD) create both quality and spatial scalable bit streams, allowing to decode animations in a progressive manner with very fine granularity.

For ensuring the streaming of the content, the FAMC encoder has been enriched with a data partitioning procedure. The coded information is structured within data packets, corresponding to disjoint temporal intervals, which are encoded independently one from another. This is equivalent to considering each temporal segment as a "mini-sequence" to be encoded in a stand-alone manner, without any reference to any other sequences.

4. COMPRESSION RESULTS

The test corpus, including about 30 animation sequences with various sizes, shapes, and motions, as well as the objective evaluation criteria has been specified within the framework of the MPEG-4 AFX Core Experiments (CE) [22] conducted by the 3DGC (*3D Graphics Compression*) subgroup of MPEG.

The comparison of compression performances with the IC approach, adopted by MPEG since 2003, showed that FAMC outperforms IC, with an average gain of 60% in bitrate [18].

Figure 3 plots the rate-distortion curve for the *Chicken* animation. The FAMC technique has been here compared to several methods of the literature: (1) TWC [7], (2) MCDWT [8], and (3) CPCA [4]. The bitrates are expressed in bits per vertex per frame (bpvf). The distortions here are expressed as the KG error [6] between initial and reconstructed meshes.

Let us note that the FAMC encoder offers the best performances with significant gains (20% - 40% in average) with regard to the state-of-the-art compresion techniques. The DCT-based FAMC version proves to be more efficient at low bitrates. The DCT+LD/LD-based FAMC versions provide a quality and spatially/temporally and spatially scalable bit stream, which offers progressive decoding and scalable rendering functionalities while ensuring competitive performances.

By using a representative subset of the MPEG test corpus, we have additionally evaluated the gain in terms of bit-rate savings obtained by using CABAC instead of a conventional N-ary arithmetic coder, when operating DCT-based FAMC at different quantization bit-depth values for the representation of prediction residuals. Corresponding results are shown in Table 1. The overall bit-rate reduction achieved by CABAC is 31% (averaged over 13 animation sequences and 6 bit depth values in the range of 4 to 14).

Because of its high compression performance and due to the large set of functionalities supported, FAMC has been recently adopted and integrated as part of the MPEG-4 Animated Framework eXtension (AFX).

5. CONCLUSION

In this paper, we have presented a novel technique for animated 3D mesh coding, so-called FAMC (*Frame-based Animated Mesh Compression*). The proposed method offers high compression rates for both geometric and photometric attributes, while supporting a complete set of advanced functionalities such as scalable rendering, progressive transmission, and streaming. The comparative experimental evaluation, carried out on the MPEG-4 test data set, objectively

Table 1. Average bit-rate savings (%) for different quantization bit-depth values obtained by comparing DCT-based FAMC using CABAC relative to the initial DCT-based FAMC approach [18] using conventional N-ary arithmetic coding.

Bit depth	4	6	8	10	12	14	4–14
Savings	24.1	34.4	36.8	35.2	30.7	25.0	**31.0**

establishes that FAMC outperforms both the IC method, previously adopted by the MPEG-4 standard (with average gains of 60%) and state-of the art techniques (20% - 40% of gains in average).

Future work will concern the design of an optimal layered decomposition by exploiting vertices coordinates additionally to mesh connectivity.

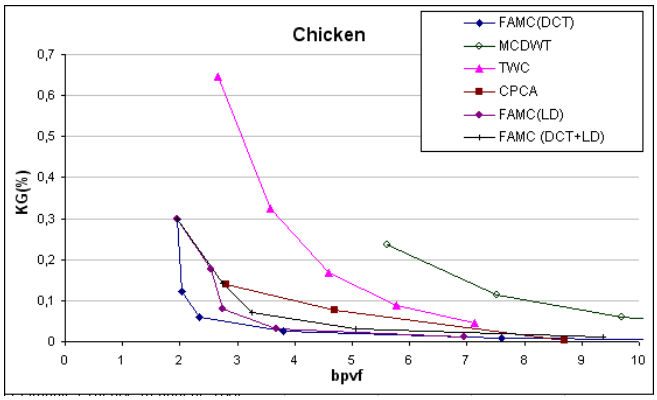

Fig. 3. FAMC vs. state of the art.

6. REFERENCES

[1] K. Mamou, T. Zaharia, and F. Prêteux, "A preliminary evaluation of 3D mesh animation coding techniques," in *SPIE Conference on Mathematical Methods in Pattern and Image Analysis*, San Diego, USA, 2005, pp. 44–55.

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

[3] E. S. Jang, J. D. K. Kim, S. Y. Jung, M. J. Han, S. O. Woo, and S. J. Lee, "Interpolator data compression for MPEG-4 animation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 989–1008, July 2004.

[4] 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.

[5] L. Váša and V. Skala, "Coddyac: Connectivity driven dynamic mesh compression," in *Proc. of the 3DTV Conference*, 2007.

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

[7] F. Payan and M. Antonini, "Temporal wavelet-based geometry coder for 3D animations," *Elsevier Computer & Graphics*, vol. 31, no. 1, pp. 78–88, 2005.

[8] 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.

[9] G. Collins and A. Hilton, "A rigid transform basis for animation compression and level of detail," in *Vision, Video, and Graphics*, Jul 2005, pp. 21–28.

[10] 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.

[11] ISO/IEC JTC1/SC29/WG11 (2007) a.k.a., "MPEG4 Part 16 AMD2: Frame-based Animated Mesh Compression," *ISO*, 2007.

[12] 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.

[13] 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.

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

[15] Khaled Mamou, Titus Zaharia, and Francoise Prêteux, "Multi-chart geometry video: A compact representation for 3D animations," in *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, May 2006, pp. 711–718.

[16] 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.

[17] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, 1998.

[18] Khaled Mamou, Titus Zaharia, Blagica Ivanova, Marius Preda, Françoise Prêteux, Benoît Meaujean, Jean Gaillard, and Olivier Marre, "Results of core experiment CE1 on mesh animation compression: skinning-based dynamic mesh compression," *ISO/IEC JTC 1/SC 29/WG 11 M14197*, 2007.

[19] D. Marpe, H. Kirchhoffer, K. Müller, and T. Wiegand, "Efficient representation and coding of prediction residuals and parameters in frame-based animated mesh compression," to be presented at IEEE Intern. Conf. on Image Processing, Oct. 2008.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B pictures," Joint Video Team, Doc. JVT-P014, Poznan, Poland, July 2005.

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

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