FRAME-BASED COMPRESSION OF ANIMATED MESHES IN MPEG-4

K. Mamou, T. Zaharia, F. Prêteux

ARTEMIS Department
Institut National des Télécommunications
Paris, France

*N. Stefanoski, J. Ostermann**

Institut für Informationsverarbeitung (TNT)
Leibniz Universität Hannover
Hannover, 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 computed from a frame-based representation and then encoded. Subsequently, either 1) DCT/lifting wavelets or 2) layer-based predictive coding is employed to exploit remaining spatio-temporal correlations in the residual signal.

The proposed encoder offers high compression performances (gains in bit rate of 60% with respect to the previous MPEG-4 technique and of 20% to 40% with respect to state-of-the-art approaches) and is well suited for compressing both geometric and photometric (normal vectors, colors...) attributes. In addition, the FAMC method 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, MPEG-4, AFX.

1. INTRODUCTION

During the last decade, animated 3D content made a spectacular break-through into the world of digital multimedia. Application domains include general public, entertainment, educational as well as professional products and services with high socio-economic impact, related to the industries of video games, CGI films, special effects, or CAD systems.

Whatever the tools and techniques used for creating content, the 3D industry privileges key-frame representations for distribution and exchange purposes. The animation is represented as a sequence of 3D key-frames, to be interpolated for ensuring the desired video framerates. The key-frame approaches lead to highly complex representations. Efficiently storing, transmitting and rendering such representations becomes then a major challenge, as testifies the rich literature dedicated to this emerging research area.

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 bandwiths, 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. Let us analyze how the above-mentioned requirements are taken into account by the state of the art techniques.

The issue of compressing dynamic 3D meshes with constant connectivity and time-varying geometry has been first considered by Lengyel in 1999 [1]. Since Lengyel pioneering work, numerous technical and methodological contributions have been proposed (see [2] for an overview) within this emerging research area. They can be structured within the following four families: (1) Local spatio-temporal predictive approaches [3, 4]; (2) Principal Component Analysis (PCA)-based techniques [5, 6, 7]; (3) Wavelet-based methods [8, 9]; and (4) Segmentation-based approaches [10, 11].

The FAMC (*Frame-based Animated Mesh Compression*) method proposed in this paper is based on a combination and extension of two approaches [12, 13]. Its core consists of a skinning model, which is exploited for motion compensation purposes, and a layered decomposition, which provides the feature of spatio-temporal scalablility. The construction of the motion model is formulated as an inverse problem: starting from an arbitrary key-frame representation, a skinning model, able to accurately describe the animation, is determined.

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 and opens perspectives of future work.

2. OVERVIEW OF THE FAMC CODER

The proposed FAMC encoder architecture is illustrated in Fig. 1. The encoder has as input a sequence of key frames (static 3D meshes) $\mathcal{F}_0, \ldots, \mathcal{F}_t, \ldots, \mathcal{F}_F$ with identical mesh connectivity. A frame $\mathcal{F}_t$ has always V vertices with 3D coordinates χ_t^v assigned to each vertex v at instant t. Additional photometric attributes like vertex normals and vertex colors can also be encoded with the FAMC encoder.

First, mesh connectivity and all 3D coordinates of $\mathcal{F}_0$ are encoded with a static mesh encoder (e.g., AFX-3DMC [14]). Subsequently the first frame is exploited in the components *Motion-model designer* and *Layered decomposition designer* in order to extract

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

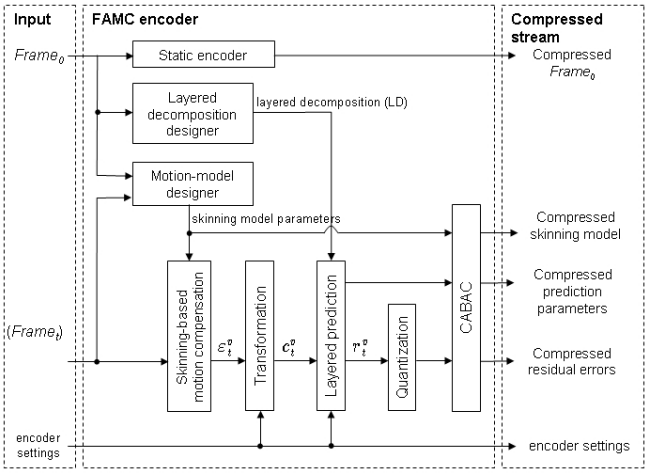

Fig. 1. Synopsis of the FAMC encoding algorithm.

information, which enables an efficient encoding of the remaining frames. Vertex coordinates (and optionally normals and colors) of frames $\mathcal{F}_1, \ldots, \mathcal{F}_F$ 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.

2.1. Skinning-based motion compensation

First, skinning model parameters are calculated in the *Motion Model Designer* module. The mesh vertices are 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.

Once the partition is determined, a skinning model is computed in the *Skinning-based motion compensation* module, and then exploited for compensating the vertex motions [12]. Thus, the predicted position $\widehat{\chi}_t^v$ of a vertex v at frame t, specified by the skinning model parameters, is given by:

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

where 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 weight vector $w^v = (w_k^v)_{k\in\{1,\ldots,K\}}$ in the L_2 sense 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 principle of linearly combining affine motions offers the advantage of obtaining a globally smooth motion field. This is in contrast with the case of using unitary-weights, generally considered by the segmentation-based approaches [10].

The motion compensation prediction errors are 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 rendering and/or smoothing 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 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 through 1D transform in temporal direction, i.e. for each v three 1D transforms are applied to $(\varepsilon_1^v, \ldots, \varepsilon_F^v)$. FAMC supports the following three transforms: (1) DCT, (2) integer-to-integer (4-2) bi-orthogonal wavelet transform implemented through lifting scheme [16], (3) bypass, i.e. $c_t^v = \varepsilon_t^v$.

2.3. Layered prediction

In this modul the predictive coding paradigm is employed in order to reduce remaining spatio-temporal dependencies between coefficients c_t^v. Already encoded coefficients of the local spatio-temporal neighborhood of c_t^v are empolyed for prediction of c_t^v. Please note that coefficients c_t^v can be either transform coefficients or prediction errors obtained after skinning-based motion compensation.

Coefficients c_t^v of an instant t are encoded layer-wise by successively encoding subsets of coefficients. For this purpose, based on mesh connectivity (*cf. Layered decomposition designer* module Fig.1) a layered decomposition (LD) is firstly determined [17, 13, 18]. A LD is specified by pairs

$$\mathcal{LD}_i^l = \left(v_i^l \,,\, \mathcal{S}_i^l \right) \quad \text{for } 1 \le i \le V_l \text{ and } 1 \le l \le L,$$

with v_i^l beeing a vertex assigned to layer l, and S_i^l beeing a set of vetices in the local neighborhood of vertex v_i^l, whose assigned coefficients are already encoded when encoding $c_t^{v_i^l}$. Now in a DPCM loop predictive coding is applied. Coefficients c_t^v assigned to time instance t are predictively encoded starting with all vertices of the base layer $l = 1$ and ending with all vertices of the highest layer $l = L$.

Without loss of generality we assume in the following that a coefficient c_t^v with assigned LD pair $(v, \mathcal{S})$ has to be predicted. In order to specify a predictor for a time instance t between three prediction modes can be selected, i.e. I-frame prediction, which exploits only encoded coefficients of the current frame for prediction, and P- and B-frame prediction, which additionally exploits encoded coefficients of one or two reference frames, respectively. The selected prediction

mode (I, P, or B) and used time instances (r_1, r_2) for reference frames are encoded for each frame as side information. Depending on the selected mode one of the following predictors is applied

$$\hat{c}_t^{v,I} = \frac{1}{|\mathcal{S}|}\sum_{v'\in\mathcal{S}} c_t^{v'},$$

$$\hat{c}_{t,r_1}^{v,P} = \mathcal{A}_t^v(c_{r_1}^v - \hat{c}_{r_1}^{v,I}) + \hat{c}_t^{v,I}, \quad \hat{c}_{t,r_1,r_2}^{v,B} = \frac{1}{2}(\hat{c}_{t,r_1}^{v,P} + \hat{c}_{t,r_2}^{v,P}).$$

I-frame prediction calculates based on the specified LD pair the average of neighboring already encoded coefficients. P-frame prediction exploits the corresponding I-frame prediction error in reference frame r_1, forming a correction vector using matrix $\mathcal{A}_t^v$. For each frame this matrix is either calculated by already encoded coefficients involving non-linear operations [13] or the identity matrix is applied leading to a linear predictor [17]. One additional bit of side information is written per frame in order to specify the matrix.

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 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, retained by the MPEG-4 / AVC - H.264 standard [19], which is well-known for its high compression performances.

The following components are encoded: (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, and prediction type (linear or non-linear), (3) residual errors, which are grouped in layers before encoding. Layer-wise encoding provides an embedded bit-stream supporting different types of scalability (*cf.* Section 3).

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* modules (LD or bypass). Different combinations of these encoder settings lead to bit streams which support different functionalities.

The *Transform* and the *Layered prediction* modul, both are organizing their output signal in layers, which induces different types of scalability. Bit streams created with setting DCT or Lift 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 setting LD creates a spatially scalable bit stream, because before encoding residuals of each frame are grouped in spatial layers, which are determined by the layered decomposition. This allows a decoding with successively increasing spatial resolution. Furthermore, the LD setting supports also temporal scalability, when frames are encoded in hierarchical B-frame order [20, 18]. 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-sequences" to be encoded in a stand-alone manner, without any reference to some 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) [21] conducted by the 3DGC (*3D Graphics Compression*) subgroup of MPEG.

The comparaison 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 [21].

Figure 2 plots the rate-distortion curves for the *Dance* and *Chicken* animations. The FAMC technique has been here compared to several methods of the literature: (1) TWC [8], (2) MCDWT [9], (3) CODDYAC [6], and (4) CPCA [5]. The bitrates are expressed in bits per vertex per frame (bpvf). The distortions here are expressed as the KG error [7] between initial and reconstructed meshes.

Let us note that in all cases and for all models 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.

Because of the high compression performances and of the complete set of functionalities supported, the FAMC has been recently adopted and integrated within the MPEG-4 Part 16 - AFX standard as Amendment 2.

4.1. Conclusion and future work

In this paper, we have presented a novel technique for animated 3D mesh coding, so-called FAMC (*Frame-based Animated Mesh Compression*). The core of the proposed method consists of (1) a skinning-based motion model, exploited in the motion compensation stage, which is automatically and optimally derived from arbitrary key-frame representations and (2) a layered decomposition, which provides the feature of spatial and temporal scalability by exploiting only mesh connectivity.

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 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).

In our future work we plan to determine optimal prediction parameters in combination with an optimal layered decomposition in order to achieve an improved compression performance, when using the encoder setting LD. I particular we plan to improve the *Layered Decomposition Designer* modul by exploiting 3D coordinates additionally to mesh connectivity.

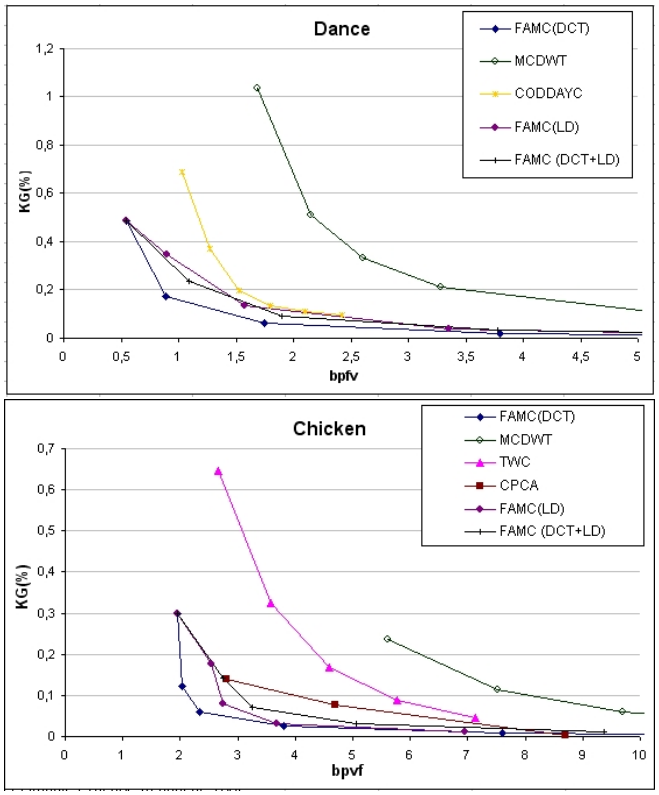

Fig. 2. FAMC vs. state of the art.

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. 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.

[3] Lawrence Ibarria and Jarek Rossignac, "Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity," in *SCA '03: Proc. of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Switzerland, Switzerland, 2003, pp. 126–135.

[4] 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.

[5] Mirko Sattler, Ralf Sarlette, and Reinhard Klein, "Simple and efficient compression of animation sequences," in *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2005, pp. 209–217, ACM Press.

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

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

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

[9] 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.

[10] 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.

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

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

[13] 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 ICIP '07 - IEEE International Conference on Image Processing*, Sep 2007.

[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] 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.

[17] 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.

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

[19] 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.

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

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