

LAYERED PREDICTIVE CODING OF TIME-CONSISTENT DYNAMIC 3D MESHES USING A NON-LINEAR PREDICTOR

Nikolče Stefanoski, Patrick Klie, Xiaoliang Liu, Jörn Ostermann (Fellow, IEEE)

Institut für Informationsverarbeitung
Leibniz Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany

ABSTRACT

We present a layered predictive compression approach for time-consistent dynamic 3D meshes. The algorithm decomposes each frame of a dynamic 3D mesh in layers employing patch-based mesh simplification techniques. This layered decomposition is consistent in time. Following the predictive coding paradigm, local temporal and spatial dependencies between layers and frames are exploited for compression. Prediction is performed vertex-wise from coarse to fine layers exploiting local linear and non-linear dependencies between vertex locations for compression. It is shown that a non-linear predictive exploitation of the proposed layered configuration of vertices can improve the compression performance upon other state-of-the-art approaches by more than 15% in domains relevant for applications.

Index Terms— Animation compression, dynamic 3D mesh coding, scalability, layered predictive coding, time-consistent mesh sequence.

1. INTRODUCTION

Applications based on static and dynamic 3D content like, 3D navigation services and 3D television, are coming closer to reality. Efficient storage and broadcasting of 3D content gets crucial importance for commercial success of this new emerging technologies. Due to an increasingly broadening range of access networks, like the Internet or local area networks, mobile networks, etc., the bit rate of compressed dynamic 3D content has to be adapted to network transfer rates and end-user devices. We developed a layered predictive compression scheme. This technique enables to encode 3D content once only, while decoding can be performed with a quality adapted to the capacity of the network and the end-user device. This is achieved by creating structured bit streams that allows for layer-wise decoding and successive reconstruction of 3D content with increasing quality.

Dynamic 3D content is usually represented by a sequence of 3D meshes called frames. Frames consist of two types of data: connectivity and 3D locations. In this paper we assume that we are dealing with frames that have constant connectivity throughout time, i.e. time-consistent mesh sequences consisting of F frames and V vertices per frame. Each vertex v in frame f is associated with a location in 3D space denoted by p_v^f for $v \in \mathcal{V} := \{1, \dots, V\}$ and $f \in \mathcal{F} := \{1, \dots, F\}$. Furthermore, the set of all vertices \mathcal{V} is decomposed into L layers, i.e. disjoint sets of vertices \mathcal{V}_l for $1 \leq l \leq L$ with the property $\cup_{l=1}^L \mathcal{V}_l = \mathcal{V}$ (see Fig. 1).

Since connectivity does not change throughout the entire mesh sequence, it has to be encoded only once. We assume that connectivity is compressed in the beginning of the encoding process by one of the nearly optimal connectivity compression techniques [1, 2]. While connectivity does not vary over time, vertices change their

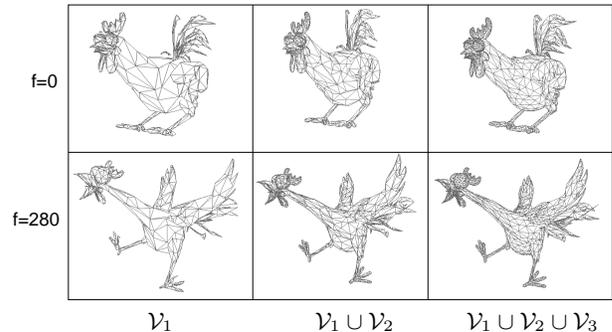


Fig. 1. Illustration of a decomposition of two frames into three layers.

location. Therefore, the major part of an encoded mesh sequence generally consists of vertex locations. For this reason, in this paper we concentrate on compression of vertex locations of time-consistent dynamic 3D meshes.

The rest of the paper is organized as follows. Section 2 gives an overview about recent developments in the area of compression of dynamic 3D meshes. An overview of the proposed scalable coder is given in Section 3, describing in detail the decomposition in layers, prediction, and entropy coding. In Section 4, compression results are evaluated and discussed. Finally we end with a conclusion in Section 5.

2. RELATED WORK

Several approaches for compression of dynamic 3D meshes have been presented recently. Karni and Gotsman [3] and Sattler et al. [4] transform dynamic meshes using principal component analysis (PCA) to reduce the amount of coded data. Guskov et al. [5] and Payan et al. [6] propose wavelet-based approaches for compression. While Guskov et al. apply the wavelet transform for each frame separately exploiting later the temporal coherence between wavelet coefficients, Payan et al. apply the wavelet transform in temporal direction on vertex trajectories and use a model-based entropy coder for entropy compression. A method for error resilient streaming of dynamic 3D meshes that minimizes the perceptual effect of data loss was introduced by Varakliotis et al. [7]. Recently Müller et al. [8] presented a rate-distortion optimized compression scheme which exploits the coherence between motion vectors by combining octree based motion vector clustering with an optimized selection of a compression mode for each cluster. Yang et al. [9] and Ibarria and Rossignac [10] presented vertex traversal based compression algorithms using linear predictors. In the first paper a parallelogram-like

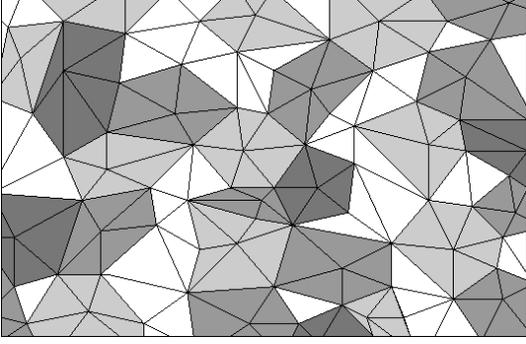


Fig. 2. Decomposition of mesh connectivity in patches with degree ≤ 6 . Gray shaded triangles are part of patches, while white triangles are not.

prediction rule is applied, while in the second paper motion vector averaging is employed to exploit local inter and intra frame coherence between vertex locations. Stefanoski and Ostermann [11] presented a non-linear angle-preserving predictor for vertex traversal based compression, improving the prediction accuracy at high bit-rates compared to linear predictors. Recently, Stefanoski et al. [12] presented a spatially and temporally scalable linear predictive coding approach, showing that a layered exploitation of spatio-temporal dependencies can improve coding efficiency. The algorithm presented in this paper is based on [11, 12]. It uses a layered configuration of vertices [12], which is exploited for predictive coding of dynamic 3D meshes with increased coding efficiency. In [12] a linear predictor is applied. Here we extend the approach by a non-linear predictor.

3. LAYERED PREDICTIVE CODING

The coder presented in this paper follows the predictive coding paradigm. In this section we describe a layered one-directional predictive coder (LOPC). A simple extension to a predictive coder supporting bi-directional prediction is described in Section 4. LOPC encodes all frames in order $1, \dots, f-1, f, \dots, F$ encoding all vertex locations p_v^{f-1} with $v \in \mathcal{V}$ before encoding all p_v^f with $v \in \mathcal{V}$. Vertex locations p_v^f of an arbitrary frame f are encoded layer-wise in order

$$v \in \mathcal{V}_1, \dots, v \in \mathcal{V}_l, \dots, v \in \mathcal{V}_L,$$

starting with all vertex locations of the base layer \mathcal{V}_1 and ending with all vertex locations of the highest layer \mathcal{V}_L . Fig. 1 illustrates the organization in layers.

3.1. Layer Design

Layers, i.e. disjoint sets \mathcal{V}_l , are defined by employing a deterministic mesh simplification technique. The basic simplification operation used in this algorithm is based on patches. A degree- d patch is a set of triangles incident to a vertex of valence d . In Fig. 3(a) a gray shaded degree-6 patch is presented. The employed simplification algorithm consists of two major steps: *Patch Decomposition* and *Simplification*.

Patch decomposition determines a series of K non-overlapping patches of degree 6 or lower, described by their middle vertices $w_1, \dots, w_k, \dots, w_K$ (s. Fig. 2). Subsequent simplification is performed only to these patches [12].

All K patches obtained during patch decomposition are traversed again in the order they were conquered during decomposition.

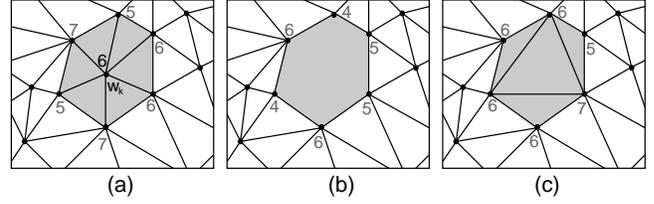


Fig. 3. Illustration of patch-based vertex removal: (a) a degree-6 patch with middle vertex w_k , (b) patch after removal of w_k , (c) re-triangulated patch. Numbers represent valences of corresponding vertices.

The middle vertex w_k of each patch is removed and the remaining polygon is re-triangulated (see Fig. 3). We define the set of all possible re-triangulations of a degree- d patch as $\mathcal{T}_d := \{1, \dots, T_d\}$ (see Fig. 4). We select that triangulation t for re-triangulating a patch which leads to the smallest deviation of patch vertex valences from the desired valence 6. Overall, this kind of selection reduces the absolute deviation of valences from valence 6. Hence, it prevents the creation of vertices with large valences, which usually lead to long narrow triangles. Furthermore, large vertex valences would also lead to many valence 3 vertices, which can not be predicted accurately later on.

3.1.1. From Simplification to Layers

The set of all vertices \mathcal{V} is decomposed in L disjoint subsets \mathcal{V}_l by recursively applying the procedures of *Patch Decomposition* and *Simplification* to the already simplified connectivity. First, patch decomposition and simplification is applied to the connectivity consisting of vertices in \mathcal{V} . Thus, a set of K_L vertices $\mathcal{V}_L = \{w_1^L, \dots, w_{K_L}^L\}$ is removed and a simplified connectivity consisting of vertices in $\mathcal{V} \setminus \mathcal{V}_L$ remains. Recursively applying this procedure to the simplified connectivity we obtain for each $l = L-1, \dots, 2$ a set of vertices $\mathcal{V}_l = \{w_1^l, \dots, w_{K_l}^l\}$ and a simplified connectivity consisting of vertices $\mathcal{V} \setminus \cup_{k=1}^L \mathcal{V}_k$. At last we define the base layer as the remaining set of vertices $\mathcal{V}_1 := \mathcal{V} \setminus \cup_{k=2}^L \mathcal{V}_k$.

Let $\mathcal{N}(v)$ denote the set neighboring vertices (1-ring) of vertex v . Patch based simplification guarantees that the neighbors of each vertex are located in the layers below, i.e. for each vertex $v \in \mathcal{V}_l$ for $l = 2, \dots, L$ we have $\mathcal{N}(v) \subset \cup_{k=1}^{l-1} \mathcal{V}_k$. In the following it will be shown that a configuration of interleaving vertices like this allows a robust exploitation of inter layer dependencies of vertex locations p_v^f in space and time.

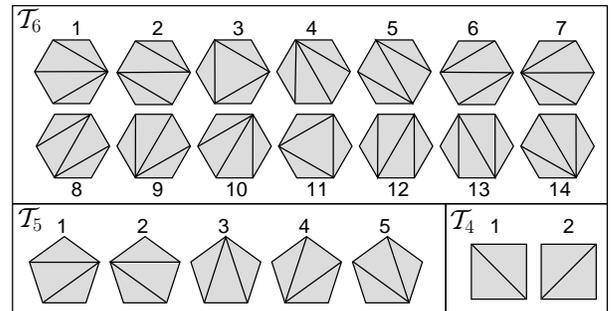


Fig. 4. Triangulations \mathcal{T}_d for $d = 4, 5, 6$.

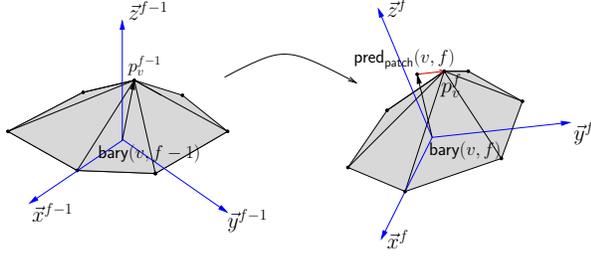


Fig. 5. Non-linear prediction based on encoded neighboring vertex locations in spatial and temporal direction.

3.2. Prediction

In order to exploit the coherence between vertex locations in spatial and temporal direction, predictive coding is applied. Vertex locations p_v^f which are part of the base layer, i.e. $v \in \mathcal{V}_1$, are encoded for $f = 1, \dots, F$ using the single resolution predictive coder [11]. After having encoded all vertex locations of the base layer of a frame f , vertex locations of higher layers of the same frame are encoded.

Vertex locations p_v^f part of a layer $l > 1$ are predicted using already encoded vertex locations. We employ two predictors, one *linear* motion vector averaging predictor $\text{pred}_{\text{mvavg}}(v, f)$ [12] and a new *non-linear* predictor $\text{pred}_{\text{patch}}(v, f)$. For each layer l in each frame f , we select that predictor, which leads to a lower prediction error. Thus, L additional bits of side information are encoded per frame, in order to specify for the decoder which predictors were used.

Both predictors, $\text{pred}_{\text{mvavg}}(v, f)$ and $\text{pred}_{\text{patch}}(v, f)$, use only encoded vertex locations of the local spatio-temporal neighborhood, i.e. p_v^{f-1} and all $p_{v'}^{f-1}$ and $p_{v'}^f$, with $v' \in \mathcal{N}(v)$ are used for prediction of p_v^f with $v \in \mathcal{V}_l$. For convenience, we define first a function which calculates the barycenter of a vertex v in frame f :

$$\text{bary}(v, f) = \frac{1}{|\mathcal{N}(v)|} \sum_{v' \in \mathcal{N}(v)} p_{v'}^f.$$

The motion vector averaging predictor also used in [9] is defined as:

$$\text{pred}_{\text{mvavg}}(v, f) = p_v^{f-1} - \text{bary}(v, f-1) + \text{bary}(v, f).$$

We propose a new *non-linear* predictor $\text{pred}_{\text{patch}}(v, f)$, which preserves the spatial relationships of vertex locations of a patch in frame $f-1$ also in frame f when predicting p_v^f (Fig. 5). Let A be an orthonormal 3×3 matrix, which is calculated based on neighboring encoded vertex locations of frames $f-1$ and f . We define

$$\text{pred}_{\text{patch}}(v, f) = A \left(p_v^{f-1} - \text{bary}(v, f-1) \right) + \text{bary}(v, f).$$

Matrix A is calculated as the product of orthogonal coordinate frames $M^{f-1} = (\vec{x}^{f-1}, \vec{y}^{f-1}, \vec{z}^{f-1})$ and $M^f = (\vec{x}^f, \vec{y}^f, \vec{z}^f)$, which are attached in the barycenters of the corresponding patches in frames $f-1$ and f , respectively (Fig. 5). Thus, $A := M^{f-1}(M^f)^T$ is a rotation matrix predicting the rotation of vector $p_v^{f-1} - \text{bary}(v, f-1)$ from frame $f-1$ to frame f relative to its barycenter. Coordinate frames M^{f-1} and M^f are determined here by calculating a regression plane through encoded neighboring vertex locations of vertex v in frame $f-1$ and frame f , respectively, which involves non-linear operations.

3.3. Quantization and Entropy Coding

Prediction errors $\delta_v^f = p_v^f - \hat{p}_v^f$ are uniformly quantized and entropy coded in order to exploit statistical dependencies. We apply an adaptive order-0 arithmetic coder combined with Golomb codes [11] for entropy coding. Separate entropy coders are employed for encoding quantized prediction errors of each layer l , adapting arithmetic coders separately to statistical distributions of residuals in each layer.

4. EVALUATION AND RESULTS

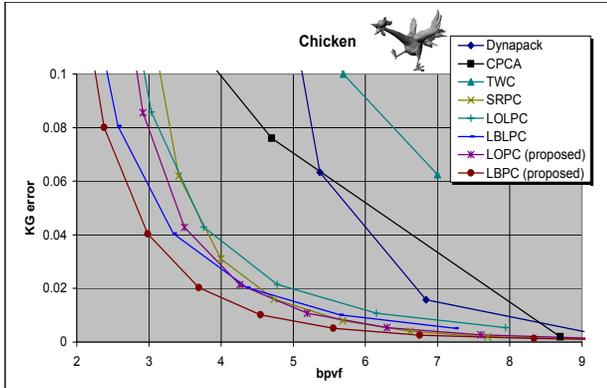
For experimental evaluation, we used the dynamic meshes *Chicken* consisting of 400 frames and 3030 vertices per frame and *Cow* consisting of 204 frames and 2904 vertices per frame. Evaluation with other dynamic meshes led to comparable results. As result of the layered representation of connectivity, we obtain time-consistent dynamic meshes in different spatial resolutions (Fig. 1). The connectivities of the sequences *Chicken* and *Cow* were decomposed in 8 and 6 layers respectively. Each time a new layer is added, the number of vertices is increased by about 38%, i.e. spatial resolution increases with a nearly constant factor.

In order to show comparative results we used the error measure presented in [3], which is denoted here as *KG error*. This measure calculates a normalized vertex-wise L_2 distance. Bit rate is measured in bits per vertex and frame (*bpvf*). In order to allow a comparison with other approaches using this measure, all vertex locations, i.e. all L layers per frame, are encoded.

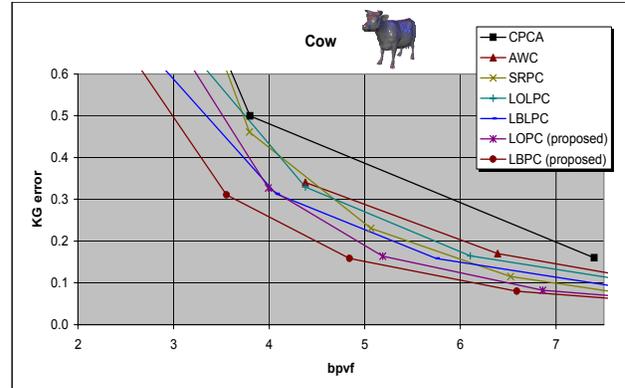
Before entropy coding, prediction errors are quantized uniformly in each spatial direction using a predefined quantization bin size Δ [11]. Operational rate-distortion curves shown in Fig. 6 were produced by varying Δ . Besides the already in Section 3 introduced LOPC, which supports only one-directional prediction based on a previous frame $f-1$, we evaluated also a coder supporting bi-directional prediction (LBPC). It is realized by first encoding an odd frame f predictively based on an encoded odd frame $f-2$ and then encoding the even frame $f-1$ using bi-directional prediction. It is performed by calculating the average of two one-directional predictions, one based on frame f and the other one based on frame $f-2$.

In Fig. 6 the proposed coders LOPC and LBPC were evaluated against state-of-the-art compression algorithms. Due to the usage of different error measures we were not able to compare against all algorithms mentioned in Section 2. We compared against the coder Dynapack [10], the single-resolution predictive coder (SRPC) of Stefanoski and Ostermann [11], the layered one- and bi-directional linear predictive coders (LOLPC and LBLPC) of Stefanoski et al. [12], the wavelet based approaches of Payan et al. (TWC) [6] and Guskov et al. (AWC) [5], and the PCA-based approach of Sattler et al. (CPCA) [4].

Both presented coders, LOPC and LBPC, outperform all other approaches. Note that an error of 0.02 for *Chicken* and 0.15 for *Cow* can be regarded as lossless with regard to visual quality. In this domain wavelet and PCA based approaches are outperformed significantly. At an error of 0.02 for *Chicken* LOPC achieved gains in bit-rate of about 5.5% against SRPC, while LBPC shows gains of over 17%. Similarly at an error of 0.15 for *Cow* LOPC achieved gains in bit-rate of about 14% against SRPC, while LBPC shows gains of over 19%. In domains of higher errors gains are even larger, i.e. over 20%. Obviously, a layered configuration of interleaving vertices increases the coding efficiency of predictive coding approaches. LOPC achieves gains in bit-rate of about 13% compared to LOLPC, while LBPC achieves even higher gains of over 15% compared to LBLPC. Thus, an additional exploitation of non-linear dependencies leads to



(a) Sequence *Chicken* encoded using $l=8$ layers. An error of 0.02 enables subjectively lossless representation.



(b) Sequence *Cow* encoded using $l=6$ layers. An error of 0.15 enables subjectively lossless representation.

Fig. 6. Evaluation results.

significant gains in domains relevant for applications. LBPC shows better compression performance than LOPC. This has to be attributed to an increased predictor robustness against quantization errors, i.e. an increased prediction accuracy, due to bi-directional prediction. Overall, layered exploitation of non-linear dependencies leads to additional gains in bit-rate providing the feature of layered decoding without any costs in bit-rate.

5. CONCLUSION

In this paper, we presented a layered predictive coder for time-consistent dynamic 3D meshes. Layers were defined by employing patch-based mesh simplification techniques and a layer-wise predictor selection was applied in order to exploit linear and non-linear spatio-temporal dependencies between layers and frames. We experimentally showed that a layered configuration of vertices improves the exploitation of spatio-temporal dependencies. The proposed algorithm outperforms state-of-the-art approaches with gains of more than 15% in domains relevant for applications. Furthermore, it can be used for layered decoding and it is applicable for real-time compression due to its low computational cost (linear run-time in the number of vertices).

6. ACKNOWLEDGMENT

This work is partly supported by the EC within FP6 under Grant 511568 with the acronym 3DTV. We would like to thank Matthias Müller from ETH Zürich for providing the *Cow* sequence. The *Chicken* character was created by Andrew Glassner, Tom McClure, Scott Benza, and Mark Van Langeveld. This short sequence of connectivity and vertex position data is distributed solely for the purpose of comparison of geometry compression techniques.

7. REFERENCES

- [1] P. Alliez and M. Desbrun, “Valence-driven connectivity encoding for 3d meshes,” *Comput. Graph. Forum*, vol. 20, no. 3, 2001.
- [2] J. Rossignac, “Edgebreaker: Connectivity compression for triangle meshes,” *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 1, pp. 47–61, 1999.
- [3] Z. Karni and C. Gotsman, “Compression of soft-body animation sequences,” *Computers & Graphics*, vol. 28, no. 1, pp. 25–34, 2004.
- [4] M. Sattler, R. Sarlette, and R. Klein, “Simple and efficient compression of animation sequences,” in *ACM/EG Symposium on Computer Animation*, NY, USA, 2005, pp. 209–217, ACM Press.
- [5] I. Guskov and A. Khodakovsky, “Wavelet compression of parametrically coherent mesh sequences,” in *Proceedings of ACM/EG Symposium on Computer Animation*, August 2004, pp. 183–192.
- [6] F. Payan and M. Antonini, “Temporal wavelet-based geometry coder for 3d animations,” *Computers & Graphics*, 2006.
- [7] S. Varakliotis, S. Hailes, and J. Ostermann, “Optimally smooth error resilient streaming of 3d wireframe animations,” in *VCIP*, 2003, pp. 1009–1022.
- [8] K. Müller, A. Smolic, M. Kautzner, P. Eisert, and T. Wiegand, “Predictive compression of dynamic 3D meshes,” in *International Conference on Image Processing*, 2005, pp. I: 621–624.
- [9] J.-H. Yang, C.-S. Kim, and S. U. Lee, “Compression of 3-d triangle mesh sequences based on vertex-wise motion vector prediction,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 12, no. 12, pp. 1178–, 2002.
- [10] L. Ibarria and J. Rossignac, “Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity,” in *Proceedings of Eurographics '03*, 2003, pp. 126–135, Eurographics Association.
- [11] N. Stefanoski and J. Ostermann, “Connectivity-guided predictive compression of dynamic 3d meshes,” in *Proceedings of the International Conference on Image Processing, ICIP2006*, Atlanta, 2006.
- [12] N. Stefanoski, X. Liu, P. Klie, and J. Ostermann, “Scalable linear predictive coding of time-consistent 3d mesh sequences,” 3DTV-CON, The True Vision - Capture, Transmission and Display of 3D Video, 2007.