CONNECTIVITY-GUIDED PREDICTIVE COMPRESSION OF DYNAMIC 3D MESHES

Nikolče Stefanoski, Jörn Ostermann

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

ABSTRACT

We introduce an efficient algorithm for real-time compression of temporally consistent dynamic 3D meshes. The algorithm uses mesh connectivity to determine the order of compression of vertex locations within a frame. Compression is performed in a frame to frame fashion using only the last decoded frame and the partly decoded current frame for prediction. Following the predictive coding paradigm, local temporal and local spatial dependencies between vertex locations are exploited. In this framework we present a novel angle preserving predictor and evaluate its performance against other state of the art predictors. It is shown that the proposed algorithm improves up to 25% upon the current state of the art for compression of temporally consistent dynamic 3D meshes.

***Index Terms*—** Mesh compression, dynamic 3D mesh coding, animation compression, prediction methods, linear predictive coding, non-linear predictive coding.

1. INTRODUCTION

Dynamic 3D meshes are more and more used to represent realistic 3D visual data. Applications can be found in computer games, character animations, avatars, physical simulations, etc. Dynamic meshes are usually represented as a series of static 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. temporally consistent dynamic meshes 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 $1 \leq v \leq V$ and $1 \leq f \leq F$. Since connectivity does not change throughout the entire mesh sequence, it has to be encoded only once. We assume that connectivity is compressed by one of the nearly optimal connectivity compression techniques [1, 2, 3, 4]. However, in contrast to connectivity, vertices change their location in time. 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 temporally consistent dynamic meshes.

Recently, several approaches for compression of dynamic 3D meshes have been presented. Karni and Gotsman [5] and Sattler et al. [6] represent dynamic meshes using principal component analysis (PCA) to reduce the amount of data. In this mesh representation, the first approach uses linear prediction to exploit remaining temporal coherence, while in the second paper mesh segmentation is applied in order to exploit the coherence of rigid body parts. Guskov et al. [7] and Payan et al. [8] 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 compression. Recently, Müller et al. [9] presented an approach which exploits the coherence of motion vectors between consecutive frames, using a clustering algorithm based on octrees. A method for error resilient streaming of dynamic 3D meshes, that minimizes the perceptual effect of data loss, was introduced by Varakliotis et al. [10]. Yang et al. [11] and Ibarria and Rossignac [12] presented vertex traversal based compression algorithms. In the first paper a parallelogram-like 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. With the exception of the two last approaches, all other approaches can not be used for real-time compression. Our algorithm is closely related to the last two approaches.

The rest of the paper is organized as follows. Section 2 gives an overview of the proposed compression algorithm, describing in detail different realizations of predictors, performed quantization, and entropy coding. In section 3, compression results are evaluated and discussed. The influence of quantization errors to the prediction error is analyzed in section 4. Finally, we end with a conclusion in section 5.

2. ALGORITHM OVERVIEW

The algorithm follows the predictive coding paradigm. All vertices are traversed and encoded in order

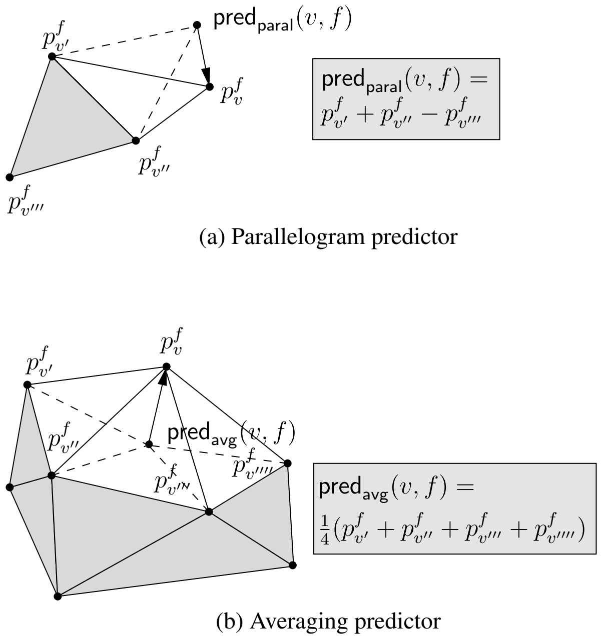
$$p_1^1, \ldots, p_V^1, \ldots, p_1^f, \ldots, p_V^f, \ldots, p_1^F, \ldots, p_V^F,$$

traversing first all vertices within frame f, before traversing vertices of the next frame $f+1$. The order of traversal and encoding within a frame f is performed in a breath-first region growing order [13], which is realized as follows. The traversal of each frame f starts with an arbitrarily chosen, initial, seed triangle PCM-encoding its incident vertex locations p_1^f, p_2^f, p_3^f. The growing region is initialized by this seed triangle. The corresponding three triangle edges are enqueued into a FIFO. Obviously, each edge in the FIFO always has two incident triangles and one of these triangles always lies in the growing region. In the traversal loop, the first edge of the FIFO is dequeued. If there is an incident triangle to this edge which is not yet part of the region, it is added to the region and its edges are enqueued into the FIFO. Every time a new vertex is encountered during traversal, its location p_v^f is predicted from so far encoded vertex locations of the current frame f and the previous frame $f-1$. Only prediction error vectors

$$\delta_v^f := p_v^f - \mathsf{pred}(v, f)$$

between the original location p_v^f and the predicted location $\mathsf{pred}(v, f)$ are encoded, after previously having quantized them. Since a previous frame for frame $f = 1$ does not exist, it is encoded using a

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


$\mathsf{pred}_{\mathsf{paral}}(v,f)$
$p_{v'}^f$
p_v^f
$p_{v''}^f$
$p_{v'''}^f$
$\mathsf{pred}_{\mathsf{paral}}(v,f) =$
$p_{v'}^f + p_{v''}^f - p_{v'''}^f$
p_v^f
$p_{v'}^f$
$p_{v''}^f$
$\mathsf{pred}_{\mathsf{avg}}(v,f)$
$p_{v''''}^f$
$p_{v'''}^f$
$\mathsf{pred}_{\mathsf{avg}}(v,f) =$
$\frac{1}{4}(p_{v'}^f + p_{v''}^f + p_{v'''}^f + p_{v''''}^f)$

(a) Parallelogram predictor

(b) Averaging predictor

Fig. 1. Linear predictors

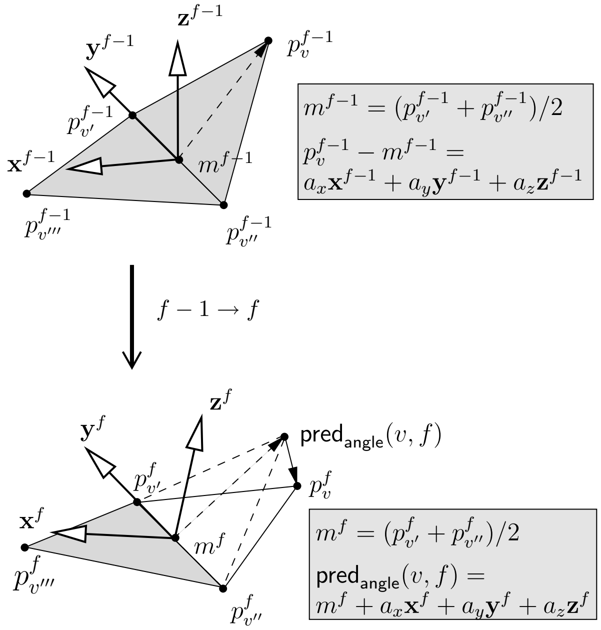

$\mathbf{z}^{f-1}$
$\mathbf{y}^{f-1}$
p_v^{f-1}
$p_{v'}^{f-1}$
$\mathbf{x}^{f-1}$
m^{f-1}
$p_{v'''}^{f-1}$
$p_{v''}^{f-1}$
$m^{f-1} = (p_{v'}^{f-1} + p_{v''}^{f-1})/2$
$p_v^{f-1} - m^{f-1} =$
$a_x\mathbf{x}^{f-1} + a_y\mathbf{y}^{f-1} + a_z\mathbf{z}^{f-1}$
$f-1 \rightarrow f$
$\mathbf{z}^f$
$\mathbf{y}^f$
$\mathsf{pred}_{\mathsf{angle}}(v,f)$
$p_{v'}^f$
p_v^f
$\mathbf{x}^f$
m^f
$p_{v'''}^f$
$p_{v''}^f$
$m^f = (p_{v'}^f + p_{v''}^f)/2$
$\mathsf{pred}_{\mathsf{angle}}(v,f) =$
$m^f + a_x\mathbf{x}^f + a_y\mathbf{y}^f + a_z\mathbf{z}^f$

Fig. 2. A non-linear angle preserving predictor

predictor which predicts p_v^1 based only on already encoded vertices of this frame. The traversal loop is iterated until the FIFO is empty and all vertices of frame f are encoded. Iterating this process for all frames f in ascending order from 1 to F, all the vertex locations p_v^f of the mesh sequence are encoded. While decoding, the same traversal and the same prediction is performed, but now the delta vectors are decoded to δ'^f_v distorted by the quantization error $\varepsilon_v^f = \delta_v^f - \delta'^f_v$. Vertex locations $p'^f_v = \mathsf{pred}(v,f) + \delta'^f_v$, distorted by quantization, are then reconstructed.

2.1. Predictors

In general, vertex locations of meshes representing real objects, show correlations. Predictors are used in order to exploit this geometric coherence between vertex locations for compression. The *parallelogram* predictor (Fig. 1a) predicts a location based on one incident triangle, which is part of the gray growing region, by creating a parallelogram. The *averaging* predictor, illustrated in Figure 1b, calculates the average of all neighboring vertex locations, which are part of the growing region. Both predictors are using already encoded vertex locations of the current frame only. Variants of these predictors are also used for compression of dynamic meshes. Ibarria and Rossignac proposed the *extended Lorenzo* predictor (ELP) for vertex traversal based compression of dynamic meshes [12]

$$\mathsf{pred}_{\mathsf{ELP}}(v,f) := \mathsf{pred}_{\mathsf{paral}}(v,f) - \mathsf{pred}_{\mathsf{paral}}(v,f-1) + p_v^{f-1}.$$

Yang et al. [11] employed a motion vector averaging predictor

$$\mathsf{pred}_{\mathsf{mvavg}}(v,f) := \mathsf{pred}_{\mathsf{avg}}(v,f) - \mathsf{pred}_{\mathsf{avg}}(v,f-1) + p_v^{f-1}.$$

These all are *linear* predictors of the type

$$\mathsf{pred}(v,f) = \sum_{(v',f')\in N(v,f)} c_{v'}^{f'}\, p_{v'}^{f'}$$

with side condition $\sum c_{v'}^{f'} = 1$ while $N(v,f)$ being the set of indices of already encoded vertex locations. In all presented predictors almost all coefficients $c_{v'}^{f'}$ are zero, since they are working in a local temporal and spatial neighborhood of vertex v in frame f.

We propose a new *non-linear* predictor with *angle preserving* properties (Fig. 2). The predictor predicts the location $\mathsf{pred}_{\mathsf{angle}}(v,f)$, preserving the angle between plane $(p_{v'}^{f-1}, p_{v''}^{f-1}, p_{v'''}^{f-1})$ and plane $(p_{v'}^{f-1}, p_{v''}^{f-1}, p_v^{f-1})$ in frame f. Calculation of $\mathsf{pred}_{\mathsf{angle}}(v,f)$ is performed using orthonormal local coordinate frames $(\mathbf{x^{f-1}}, \mathbf{y^{f-1}}, \mathbf{z^{f-1}})$ and $(\mathbf{x^f}, \mathbf{y^f}, \mathbf{z^f})$ in frame $f-1$ and f, respectively. These are attached in the corresponding edge centers m^{f-1} and m^f. Their $\mathbf{y}$-axis is aligned with edge $(p_{v'}^{f-1}, p_{v''}^{f-1})$ and $(p_{v'}^f, p_{v''}^f)$, respectively. The corresponding $\mathbf{x}$-axis is lying in the plane of the supporting triangle. Prediction is based on the coordinates $(a_x, a_y, a_z)^T$, which represent point $p_v^{f-1} - m^{f-1}$ relative to basis $(\mathbf{x^{f-1}}, \mathbf{y^{f-1}}, \mathbf{z^{f-1}})$ (see Fig. 2). Thus, $\mathsf{pred}_{\mathsf{angle}}(v,f)$ is a perfect predictor if only rigid motion is carried out from frame to frame, otherwise it preserves the angle between planes.

2.2. Quantization

Prediction error vectors δ_v^f are quantized before entropy coding in order to exploit statistical dependencies. This induces an irreversible loss of information. We quantize each component of $\delta_v^f = (\delta_{v,1}^f, \delta_{v,2}^f, \delta_{v,3}^f)^T$ uniformly using a quantization bin with width Δ, which controls the amount of information loss. Quantization is performed for each component separately, according to

$$q\left(\delta_{v,*}^f\right) := \left\lfloor \delta_{v,*}^f/\Delta + \frac{1}{2} \right\rfloor = i_{v,*}^f$$

for $* \in \{1,2,3\}$. The inverse quantisation

$$r\left(i_{v,*}^f\right) := i_{v,*}^f \cdot \Delta = \delta'^f_{v,*}$$

reconstructs distorted prediction error vectors $\delta'^f_{v,*}$.

2.3. Entropy Coding

We apply an adaptive order-0 arithmetic coder [14, 15] combined with Golomb codes [15], in order to encode the quantized residuals $i^f_{v,*}$. We encode all residuals ranging between $-3, \dots, 3$ using arithmetic codes, adapting the probabilities to the residuals of the previous frame. All values outside of this range are encoded using Golomb codes, which are optimal Huffman codes for discrete probability distributions with exponential decay.

3. EXPERIMENTAL RESULTS AND DISCUSSION

For experimental evaluation, we used the mesh sequence *Chicken Crossing*, consisting of 400 frames and 3030 vertices per frame. Evaluation with other mesh sequences led to comparable results. The evaluation was performed using a normalized vertex-wise L_2 norm, denoted here by d_a, which was proposed in [5]. Bit rate is given in bits per vertex and frame (bpvf). We apply the same Δ for quantization during compression, distributing the quantization error uniformly throughout the mesh sequence.

In Fig. 3a, rate-distortion curves produced by varying Δ using different realizations of predictors, are shown. Besides already presented predictors, we also evaluated a combined predictor (angle + mvavg). This predictor is realized by selecting for each frame separately one of the predictors of $\mathsf{pred}_{\mathsf{angle}}(v, f)$ or $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$. The predictor leading to a lower prediction error is selected for compression. One additional bit of side information is encoded per frame, in order to specify to the decoder which predictor is used.

In the area of bit rates over 6.4 bpvf, the angle preserving predictor achieves significant gains against the linear predictors $\mathsf{pred}_{\mathsf{ELP}}(v, f)$ and $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$. This is because $\mathsf{pred}_{\mathsf{angle}}(v, f)$ exploits not only linear dependencies, but also dependencies of higher order. Below 6.4 bpvf, the performance of this predictor rapidly drops, due to the dominating influence of the quantization error on the bit rate. $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$ achieves significant gains at low bit rates because of its property to reduce the influence of the quantization error through averaging. The combined predictor shows the best performance despite the additional side information. It prefers $\mathsf{pred}_{\mathsf{angle}}(v, f)$ at bit rates over 6.4 bpvf, while $\mathsf{pred}_{\mathsf{angle}}(v, f)$ is more often applied at lower bit rates.

In Fig. 3b, the proposed algorithm using the combined predictor is evaluated against other 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 the introduction. We compared against the algorithm Dynapack [12], which uses the predictor $\mathsf{pred}_{\mathsf{ELP}}(v, f)$, and the recently presented approaches of Payan et al. [8] and Sattler et al. [6], based on wavelets and PCA. We achieved significant gains in bit rate, especially in the area of errors below 0.03. Here we achieved gains of over 2 bpvf or over 25%. Obviously, exploitation of local spatio-temporal coherence of vertex locations using predictors of higher order can lead to significant gains, in comparison to approaches exploiting long term dependencies within vertex trajectories or aproaches based on PCA. Furthermore, most of the approaches mentioned in the introduction are computationally more demanding than our algorithm.

4. ANALYSIS

Our experiments showed that the probability distribution of $\delta^f_{v,*}$ can be nicely fitted by a Laplacian distribution with zero mean. Hence, the associated mean squared quantization error $e^2_{q,*} = E[(\varepsilon^f_{v,*})^2|\delta'^f_{v,*}]$ is independent of $\delta'^f_{v,*}$, since quantization is performed uniformly. Furthermore, we verified by experiment, that the resulting quantization error $\varepsilon^f_{v,*}$ can be regarded as a stationary, zero-mean white-noise process with autocorrelation function

$$E\left[\varepsilon^f_{v,*} \cdot \varepsilon^{f-j}_{v-i,*-k}\right] = \begin{cases} e^2_{q,*} & : \; i = j = k = 0 \\ 0 & : \; \text{else} \end{cases} .$$

Therefore, the reconstructed vertex locations at the decoder side $p'^f_v = p^f_v + \varepsilon^f_v$ can be considered statistically as original vertex locations p^f_v distorted with additive, stationary, zero-mean white-noise $\varepsilon^f_v = (\varepsilon^f_{v,1}, \varepsilon^f_{v,2}, \varepsilon^f_{v,3})^T$.

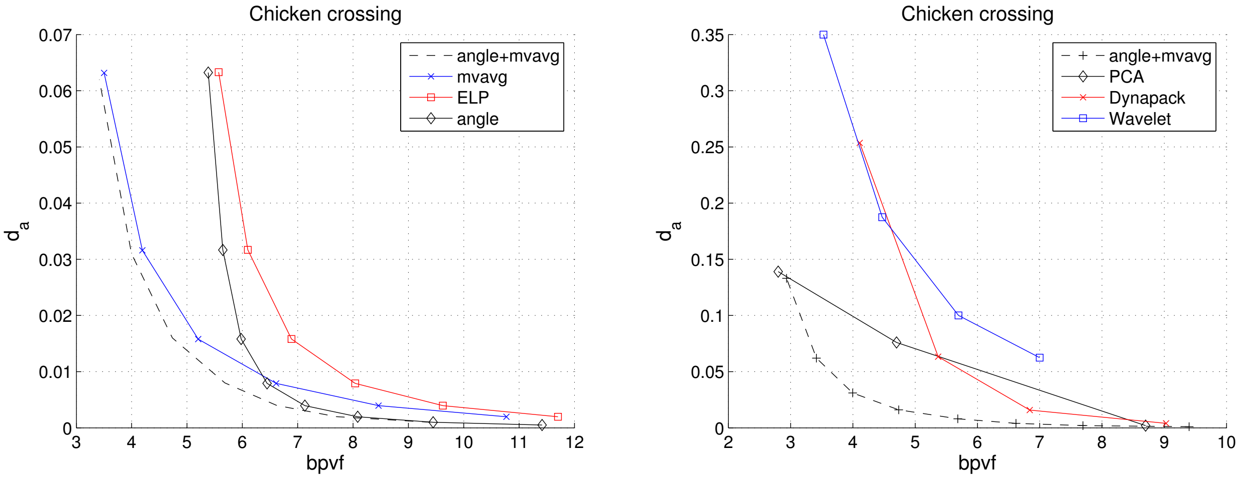
The prediction error δ^f_v is calculated using a predictor which works on the already reconstructed vertex location $p'^{f'}_{v'}$. Having this in mind, we are evaluating in particular the influence of the quantization error ε^f_v to the prediction error δ^f_v. Without loss of generality we calculate the mean squared prediction error of the first component $\delta^f_{v,1} = p^f_{v,1} - \mathsf{pred}(v, f)_1$ for a linear predictor:

$$\begin{aligned}
\mathcal{E} = E\left[\left(\delta^f_{v,1}\right)^2\right] &= E\left[\left(p^f_{v,1} - \sum_{(v',f')\in N(v,f)} c^{f'}_{v'} p'^{f'}_{v',1}\right)^2\right] \\
&= E\left[\left(p^f_{v,1} - \sum_{(v',f')\in N(v,f)} c^{f'}_{v'} (p^{f'}_{v',1} + \varepsilon^{f'}_{v',1})\right)^2\right] \\
&= E\left[\left(p^f_{v,1} - \sum_{(v',f')\in N(v,f)} c^{f'}_{v'} p^{f'}_{v',1}\right)^2\right] \\
&\quad + e^2_{q,1} \sum_{(v',f')\in N(v,f)} (c^{f'}_{v'})^2 \\
&= \mathcal{E}^p + \mathcal{E}^q .
\end{aligned}$$

The overall mean squared prediction error $\mathcal{E}$ is composed of two types of errors. The first error $\mathcal{E}^p$ depends on the original vertex locations, while the latter error $\mathcal{E}^q$ depends on the mean squared quantization error. Both errors are depending on the coefficients $c^{f'}_{v'}$ of the used predictor. A careful selection of a predictor can reduce the overall error $\mathcal{E}$. The suitability of a predictor depends on one hand on its prediction accuracy based on original vertex locations ($\mathcal{E}^p$) and on the other hand on the applied level of quantization ($\mathcal{E}^q$). Obviously, $\mathcal{E}^q$ dominates the overall error $\mathcal{E}$ when performing coarse quantization. We obtain $\mathcal{E}^q_{\mathsf{ELP}} = 7\, e^2_{q,1}$ when using the predictor $\mathsf{pred}_{\mathsf{ELP}}(v, f)$ for compression, while when using $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$ we get $\mathcal{E}^q_{\mathsf{mvavg}} < 2\, e^2_{q,1}$. Hence, the mean squared quantization error $e^2_{q,1}$ is more amplified by $\mathsf{pred}_{\mathsf{ELP}}(v, f)$ than by $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$. This demonstrates the superiority of $\mathsf{pred}_{\mathsf{mvavg}}(v, f)$ over $\mathsf{pred}_{\mathsf{ELP}}(v, f)$ when performing coarse quantization (cp. Fig. 3a). We showed experimentally in section 3 that the combination of motion vector averaging and non-linear angle preserving prediction can reduce the overall error, leading even more to improved compression results.

5. CONCLUSION

In this paper, we presented a vertex traversal based compression algorithm for compression of temporally consistent dynamic 3D meshes. We combined breath-first region growing vertex traversal with a novel non-linear angle preserving predictor, exploiting local spatio-temporal coherence between vertex locations. The advantage of this approach is that due to its low computational cost (linear runtime in



(a) Comparison of predictors

(b) Comparison of compression algorithms

Fig. 3. Evaluation results

the number of vertices), it can be used for real-time compression. Furthermore, we experimentally showed that significant compression gains (over 25%) can be achieved, using a combination of linear and non-linear predictors. We showed that the proposed compression approach outperformes state-of-the-art compression methods based on wavelets or PCA.

One drawback of the proposed algorithm is its limited compression performance at very low bit rates (below 3 bpvf). This is because of the dominating influence of the quantization error to the overall prediction error, when performing coarse quantization (see section 4).

6. ACKNOWLEDGMENTS

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] S. Gumhold and W. Strasser, "Real time compression of triangle mesh connectivity," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 1998, pp. 133–140, ACM Press.

[2] C. Touma and C. Gotsman, "Triangle mesh compression.," in *Graphics Interface*, 1998, pp. 26–34.

[3] J. Rossignac, "Edgebreaker: Connectivity compression for triangle meshes.," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 1, pp. 47–61, 1999.

[4] P. Alliez and M. Desbrun, "Valence-driven connectivity encoding for 3d meshes.," *Comput. Graph. Forum*, vol. 20, no. 3, 2001.

[5] Z. Karni and C. Gotsman, "Compression of soft-body animation sequences," *Computers & Graphics*, vol. 28, no. 1, pp. 25–34, 2004.

[6] M. Sattler, R. Sarlette, and R. Klein, "Simple and efficient compression of animation sequences," in *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, New York, NY, USA, 2005, pp. 209–217, ACM Press.

[7] I. Guskov and A. Khodakovsky, "Wavelet compression of parametrically coherent mesh sequences," pp. 183–192.

[8] F. Payan and M. Antonini, "Wavelet-based compression of 3d mesh sequences," in *Proceedings of IEEE ACIDCA-ICMI'2005*, Tozeur, Tunisia, november 2005.

[9] K. Muller, 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.

[10] Socrates Varakliotis, Stephen Hailes, and Jörn Ostermann, "Optimally smooth error resilient streaming of 3d wireframe animations.," in *VCIP*, 2003, pp. 1009–1022.

[11] 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.

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

[13] S. Gumhold and R. Amjoun, "Higher order prediction for geometry compression," in *SMI '03: Proceedings of the Shape Modeling International 2003*. 2003, p. 286, IEEE Computer Society.

[14] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inf. Syst.*, vol. 16, no. 3, pp. 256–294, 1998.

[15] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, 1996.