SCAN-BASED COMPRESSION OF 3D MESH SEQUENCES WITH GEOMETRY COMPENSATION

Yasmine Boulfani and Marc Antonini

Laboratoire I3S - UMR 6070 CNRS , Université de Nice - Sophia Antipolis
Route des Lucioles - 06903 Sophia Antipolis FRANCE
phone: +33 4 92 94 27 22, fax: +33 4 92 94 28 98, email: {boulfani,am}@i3s.unice.fr

ABSTRACT

We introduce in this paper a new compression process to encode the geometry of 3D mesh sequences (with a fixed connectivity). The proposed method uses a scan-based wavelet transform with an original approach which consists to compensate the geometry displacements of the mesh sequence. The proposed coder is based on temporal lifting scheme and bit allocation using statistical models for the probability densities of the temporal wavelet coefficients. Then, scalar quantization and entropy coding are used to encode the quantized subbands and the geometry displacement vectors. The resulting compression scheme shows that good coding performances are obtained when the compensation of the geometry is done on the whole sequence, which requires large memory. Furthermore, we showed that when low memory scan-based compression is done, the performances of the encoder approach the ones obtained when the whole sequence is known. Also, simulation results show that the proposed algorithm provides better compression performances than some state of the art coders.

1. INTRODUCTION

The continuous development of the multimedia hardware and software causes an increasing interest for the use of 3D contents which permit a three-dimensional modeling of the real world. Although animated sequence of 3D meshes may be produced and represented in a variety of ways, they are often stored and transmitted as series of consecutive triangular meshes generally called frames. Each frame is defined by the location of the vertices (geometry) and by a triangle/vertex incidence graph (connectivity). In general, the meshes are irregular, and the connectivity of the meshes may also evolve with time. Nevertheless, in this paper, we restrict our attention to a class of animations in which the connectivity is identical in all frames.

Similarly to several techniques [1, 3, 4, 5], we propose an efficient way to compress animated sequences by considering the sequence as geometric deformations of the geometry of the first frame. Among these techniques, Alexa and Müller [1] , proposed a coding scheme based on the Principal Component Analysis (PCA). Karni and Gotsman improved this method by further exploiting the spatial and temporal coherence and finally encode the principal components with a predictive coding scheme called LPC [5]. Also, Ibarria and Rossignac proposed a method which predicts the position of a vertex by using the already decoded vertices of its spatial neighbourhood and its position in previous frames[4]. Finally, Guskov and Khodakovsky presented recently a compression algorithm based on a multiresolution analysis [3]. In their work the analysis is applied on the geometry of each frame to obtain a multiresolution representation. A predictive coding scheme is then applied on the resulting details of each frame.

In this paper, we propose an alternative compression method using scan-based compression of 3D mesh sequences with geometry displacement compensation. More precisely, a scan-based estimation of the geometry displacement is done providing displacement vectors. These vectors are then used for the geometry displacement compensation before applying the temporal wavelet transform. To reduce their cost, the displacement vectors are entropy coded.

A scan-based temporal wavelet decomposition is applied on the compensated mesh sequence to get different temporal subbands which are encoded with a model-based bit allocation proposed in [8]. This scan-based compression method requires low memory and allows to approach the coding performances when the whole sequence can be stored in memory.

The rest of this paper is organized as follows. Section 2 describes the structure of the proposed compression scheme. Section 3 deals with the scan-based wavelet transform using the geometry compensation. Section 4 presents the allocation process included in the proposed coder. Experimental results are given and compared to results of several state of the art methods in Section 5. We finally conclude and propose future works in section 6.

2. COMPRESSION SCHEME

Figure 1 shows the global structure of the proposed compression algorithm.

The main steps of the algorithm are the following:

- **Compensation of the geometry and estimation of the displacement vectors**: This is an essential step of the proposed coding scheme. Contrary to most of video coders, we do not have to use a motion estimation algorithm to match the current vertices with vertices of previous frames before applying the temporal wavelet transform [2]. Here, the processed data are geometric positions in space, and as the connectivity is the same for each frame involving a fixed number of vertices along the sequence, the motion of each vertex is implicit. The estimation of the geometry displacement can then be computed analytically providing displacement vectors. This process is detailled in Section 3.
- **Temporal wavelet transform**: A decomposition on several levels allows to get the *low-frequency (LF)* subband, which represents a coarse version of the compensated sequence, and sequences of wavelet coefficients which represent the *high frequency (HF)* details. The retained

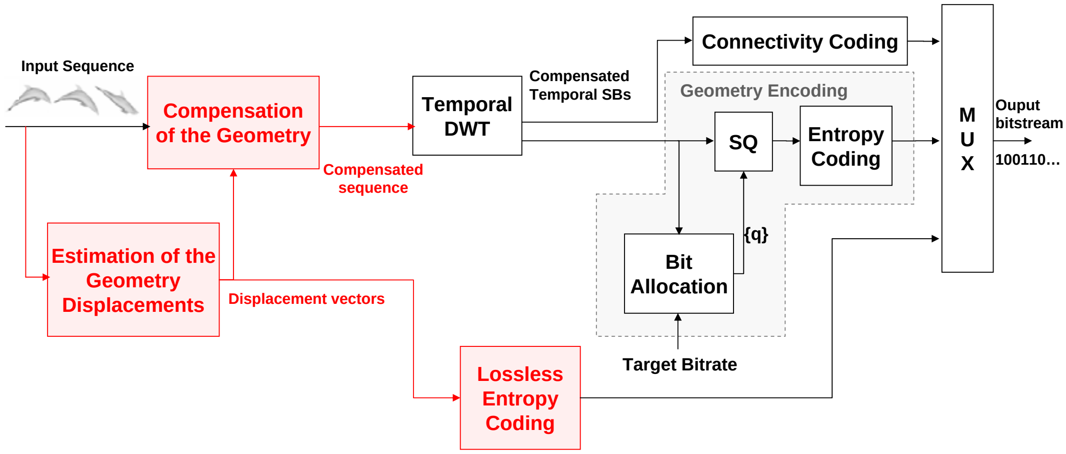

Figure 1: General structure of the proposed compression algorithm.

method is an original approach that we have proposed in [8] and [10], based on a temporal lifting scheme, in order to exploit the high temporal correlation of the geometry of successive frames. The idea was to exploit a monodimensional lifting scheme by performing the time-filtering not in the same position for all the considered frame, but by ”following the vertex in its displacement. In the rest of the paper, we use the lifting scheme $(2,0)$ presented in [8] and [10].

- **Bit allocation and scalar quantization (SQ)**: Each obtained wavelet coefficient sequence is separated in 3 subbands of coordinates that will be treated separately in the rest of the algorithm. The goal of an allocation algorithm is to find the quantizer parameters of all the subbands which allow to get the best global performances and by optimizing the trade-off between the visual quality of the reconstructed sequence and the total bitrate. The allocation process proposed here dispatches the bit budget across the wavelet sequences according to their influence on the quality of the reconstructed mesh sequence for one specific bitrate. The quantization is performed by uniform scalar quantizers.
- **Entropy coding**: Once quantized, the wavelet coefficients are entropy coded to produce the bitstream. We use a simplified version of the context-based arithmetic coder of [6]. However, the vectors of the geometry displacement are supposed to be losslessly entropy encoded and transmitted to the decoder.
- **Connectivity coding**: In order to be able to reconstruct the quantized mesh sequence after decoding, the connectivity of the original mesh sequence must also be encoded and transmitted. As the connectivity remains the same for every frame of the mesh sequence, we simply encode the connectivity of the first frame with the efficient coder of Touma and Gotsman [11].

3. SCAN-BASED GEOMETRY COMPENSATED LIFTED WAVELET TRANSFORM

A practical problem in temporal wavelet transform implementation is related to memory requirements of time filters. Usually, wavelet transform filtering is implemented by loading all the data in memory and then performing wavelet transform filtering. In the case of temporal filtering of mesh sequences, this would require a huge memory size and moreover could imply an encoding delay as long as the sequence duration itself. A simple solution to the temporal filtering problem is to crop the input sequence in several short subsequences called GOF (Groups Of Frames) and than compute temporal wavelet transform on them.

However, instead of applying the wavelet transform directly on the original sequence, we introduce a scan-based geometry compensation of the original sequence in order to reduce the energy of the wavelet coefficient.

3.1 Proposed approach for the geometry compensation

The compensation approach consists in displacing all the frames of the same GOF on the first frame of this GOF, while keeping this first frame unchanged. However, we are going to see thereafter (Section 3.2) that in order to compute the scan-based wavelet transform for a GOF, we need to use the first frame of the next GOF. So, in this case, we displace this first frame of the next GOF on the last frame of the treated GOF.

To do this geometry displacement compensation, we need a suitable set of vectors representing the geometry displacement of each frame of the GOF. The principle of the estimation of the displacement vectors is the following. let us consider the first GOF of the original sequence, $\mathscr{F}_1 = \{f_1, f_2, ..., f_k, ... f_T\}$, composed by T static meshes (frames), where each frame f_k is defined by its geometry, i.e., a set of $\mathscr{V}$ vertices and a list of triangles describing the connectivity of the vertices. However, in this paper we consider only the mesh sequences with fixed connectivity. Consequently, the list of triangles remains identical in all frames.

For each frame f_k, we associate a displacement vector field of the frame geometry called $V_{k\to1}$ computed as it is shown in the Figure 2.

For each vertex coordinate i of each frame f_k, we can compute a displacement vector of the geometry as follows:

$$V_{k\to1}(i) = f_k(i) - f_1(i), \text{ for } k = 2,3,..T \qquad (1)$$

However, for the first frame f_{T+1} of the next GOF, the displacement vector for each i is computed as:

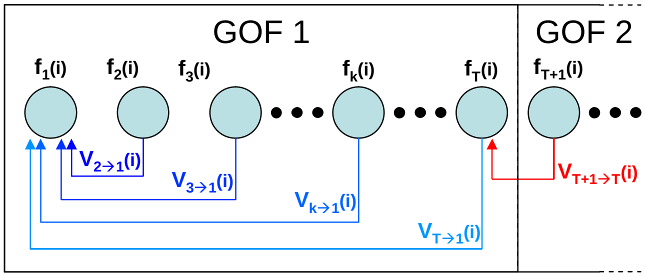

Figure 2: Estimation of the geometry displacements

$$V_{T+1\to T}(i) = f_{T+1}(i) - f_T(i) \tag{2}$$

We obtain thus $\mathscr{V}$ displacement vectors for each frame of the GOF.

In order to reduce the cost of displacement vectors, we propose to reduce their number by splitting each frame in N clusters ($\mathscr{C}_N$), and then to compute an average displacement vector in each cluster $\mathscr{C}_N$. Therefore, we will have N displacement vectors for each frame of the sequence.

The new vector for each frame and for each cluster, can be written as:

$$\bar{V}_{k\to 1}^{\mathscr{C}_N} = \frac{1}{|\mathscr{C}_N|} \sum_{i\in\mathscr{C}_N} V_{k\to 1}(i), \text{for all } N,\, k = 2,3,\ldots,T \tag{3}$$

and $|.|$ corresponds to the cardinal of $\mathscr{C}_N$.

For the first frame f_{T+1} of the next GOF, we have:

$$\bar{V}_{T+1\to T}^{\mathscr{C}_N} = \frac{1}{|\mathscr{C}_N|} \sum_{i\in\mathscr{C}_N} V_{T+1\to T}(i), \text{for all } N. \tag{4}$$

Once the suitable displacement vectors of the geometry are computed for each frame of the original GOF, we can apply the geometry compensation on this GOF and on the first frame of the next GOF. Let us denote the new GOF and the new first frame of the next GOF by $\hat{\mathscr{F}}_1 = \{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_k, \ldots \hat{f}_T\}$, and $\hat{f}_{T+1}$, respectively. Each frame is computed for each i and for each N as follows:

$$\hat{f}_k(i) = f_k(i) - \bar{V}_{k\to 1}^{\mathscr{C}_N}, \text{ when } k = 2,3,\ldots T \tag{5}$$

$$\text{and } \hat{f}_{T+1}(i) = f_{T+1}(i) - \bar{V}_{T+1\to T}^{\mathscr{C}_N} \tag{6}$$

3.2 Scan-based wavelet transform based on geometry compensation

Once each GOF of the original sequence is geometry compensated, we can apply the scan-based wavelet transform on the compensated sequence.

The Figure 3 shows the principle of the scan-based approach on the "Face" sequence by using the lifting scheme $(2,0)$ and for a GOF of 4 frames. Each GOF is independently filtered. As we mentioned in section 3.1, in order to solve the boundary problem, we need to consider the first frame of next GOF to compute the filtering of the last frame of previous GOF. The scan-based approach allows to overcome this problem, by computing the temporal transform as it would be by considering all the sequence as a whole, but

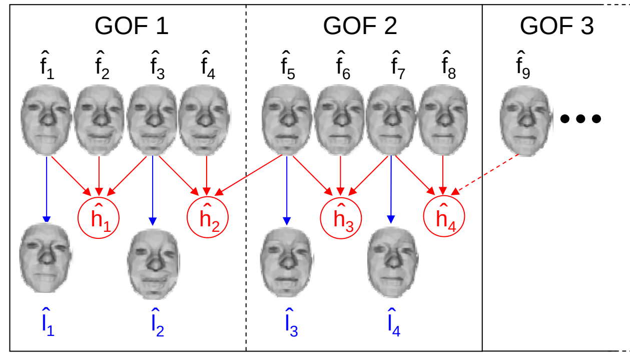

Figure 3: Example of a scan-based wavelet transform decomposition (lifting scheme $(2,0)$ - 1 level decomposition)

by keeping in memory only data necessary to compute the wavelet transform on one GOF.

Also, in order to compute the last hight frequency coefficient of the last GOF of the sequence, we do a symmetrical extension of the sequence, obtained by the replication of the next-to-last frame of the sequence.

4. BIT ALLOCATION

The *HF* details and the *LF* sequence of each GOF obtained by the scan-based temporal wavelet transform, need to be encoded. However, to optimize the quantization process, we introduce a bit allocation in the compression algorithm. The objective of an allocation process is to optimize the trade-off between rate and distortion (after lossy compression and decompression). In these works, the lifting scheme transforms the original sequence in several detail sequences. So, our allocation process aims to dispatch the bit budget across the wavelet sequences according to their influence on the quality of the quantization of the mesh sequence for one specific bitrate.

More precisely, the allocation process presented here allows to compute the set of optimal quantizers $\{q^*\}$, which minimizes the reconstructed mean square error D_T for one specific user-given target bitrate R_{target}. The solutions $\{q^*\}$ are obtained by solving the problem

$$(\mathscr{P}) \begin{cases} \text{minimize} & D_T(\{q\}) \\ \text{with constraint} & R_T(\{q\}) = R_{target}, \end{cases} \tag{7}$$

with R_T the total bitrate.

By using a lagrangian approach, the constrained allocation problem $\mathscr{P}$ can be solved by minimizing the criterion

$$J_\lambda(\{q\}) = D_T(\{q\}) + \lambda\left(R_T(\{q\}) - R_{target}\right), \tag{8}$$

with λ the lagrangian operator.

The optimal quantization steps $\{q^*\}$ are obtained by solving the following system [7]:

$$\begin{cases} \frac{\partial J_\lambda(\{q\})}{\partial q} = 0 \\ \frac{\partial J_\lambda(\{q\})}{\partial \lambda} = 0 \end{cases} \tag{9}$$

For more details about the resolution of this problem, see [7, 9].

4.1 Encoding of the LF sequence

As we stated previously, the distributions of the probability density function (pdf) of each HF coordinate set can be modeled by a *GGD*, but concerning the *LF* sequence, it is not the case. To overcome this problem, we use a *differential coding*. The main idea of such a coding is to encode the differences between the samples instead of the samples themselves. By using this coding, the pdf of the *LF* data can be finally modeled by a *GGD* [7].

In this paper, we propose a *geometric differential coding*. The idea is to process, for each frame t, the differences between coordinates in function of the vertex indexes. So, we exploit the spatial correlation existing between neighbour vertices.

$$[V(1,t)-V(0,t), V(2,t)-V(1,t), \ldots, V(\mathscr{V})-V(\mathscr{V}-1,t)],$$

with $\mathscr{V}$ the number of vertices.

5. SIMULATION RESULTS

To show the efficiency of the proposed scan-based compression algorithm with geometry compensation, we have tested its performances on different sequences. Here, we present two animation sequences: “Chicken” (2916 vertices, 5454 faces and 384 frames) and “Dolphin” (6179 vertices, 12337 faces and 64 frames). For these examples, a 4-level decomposition is used, and we test our algorithm for several size of GOF and on the whole sequence by varying the number of displacement vectors in each frame ($N = 1$ and $N = 4$).

To evaluate the quality between the original sequences and the reconstructed ones, we use the metric error called *KG error*, introduced by Karni and Gotsman in [5]. This metric corresponds to the relative discrete L_2-norm both in time and space and is expressed in percent. It is given by:

$$KG\ error = 100\frac{||G-\hat{G}||}{||G-E(G)||}$$

where G is a matrix of dimension $(3v, T)$ containing the geometry of the original sequence, $\hat{G}$ the quantized version of the geometry, and $E(G)$ an average matrix in which the t^{th} column is defined by:

$$(\overline{X_t}\,(1\ \ldots\ 1)\,, \overline{Y_t}\,(1\ \ldots\ 1)\,, \overline{Z_t}\,(1\ \ldots\ 1))^T$$

with $\overline{X_t}$, $\overline{Y_t}$, and $\overline{Z_t}$ the mean values of the coordinate sets of each frame t.

Figure 4 shows the curves *KG Error/bitrate* for the “Chicken” sequence when we use the coder presented in [8] with and without geometry compensation on the whole sequence (GOF=384 frames) and by using the proposed scan-based approach (GOF=32 frames). The bitrate is given in bits per vertex per frame.

Globally, we observe that the proposed coder with geometry compensation applied on the whole sequence (GOP=384 frames) provides the best coding performances compared to all the others, but this method requires a huge memory and moreover could imply an encoding delay as long as the sequence duration itself. Whereas, when we introduce the scan-based compression, for a GOF=32 frames, the coder using the geometry compensation presents better performances than the coder which do not use the geometry compensation.

Also, concerning the scan-based approach with geometry compensation, if we use 4 displacement vectors per frame we obtain better results than if 1 displacement vector was used. The more we introduce clusters, the better is the estimation of the geometry displacement. However this improvement is done at the cost of an increasing number of bits to describe the displacement vectors. We have thus to optimize this trade-off by finding the best configuration.

Furthermore, the proposed scan-based compression method requires low memory and allows to approach the coding performances when the whole sequence can be stored in memory.

Figures 5 and 6 compare the coding performances of the proposed coder, for “Chicken” and “Dolphin” sequences, with the static mesh coder of Touma and Gotsman [11] denoted by TG, and the PCA-based coder of Alexa and Müller denoted by *PCA* [1]. For both sequences, we observe that the proposed compression algorithm using scan-based geometry compensation is more efficient than the TG and *PCA* methods.

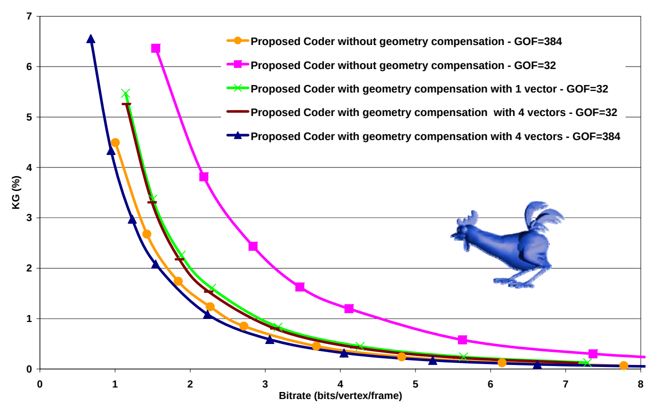

Figure 4: *KG Error/bitrate* for “Chicken” according to different proposed technics.

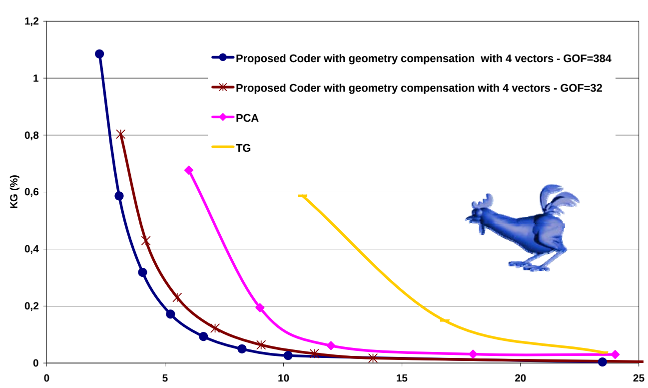

Figure 5: *KG Error/bitrate* for “Chicken” relative to different compression methods .

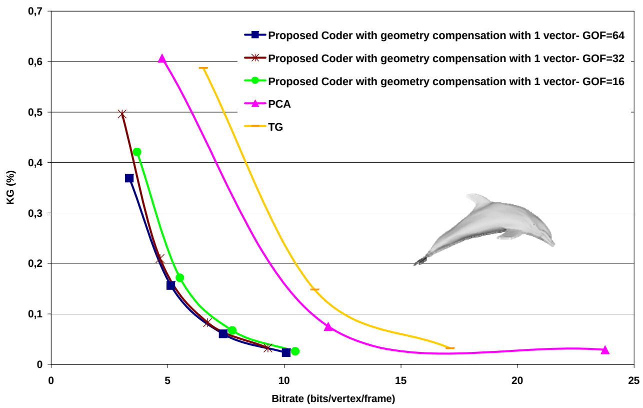

Figure 6: *KG Error/bitrate* for “Dolphin” relative to different compression methods .

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a new compression scheme for animated mesh sequences, for which the connectivity is identical in all frames. This compression scheme uses a scan-based wavelet transform with geometry compensation, which requires low memory. Experimentally, we have shown that the proposed coder using compensated geometry provides better performances than the proposed coder without using geometry compensation. Also, our scan-based coder with geometry compensation provides better performances than some state of the art coders for “Chicken” and “Dolphin” sequences .

In our current works, we are improving the geometry compensation approach.

REFERENCES

[1] M. Alexa and W. Muller. Representing animations by principal components. *Comput. Graph. Forum 19*, 3, 2000.

[2] M. Cagnazzo, T. André, M. Antonini, and M. Barlaud. A model-based motion compensated video coder with jpeg2000 compatibility. In *Proc. IEEE International Workshop on Multimedia Signal Processing*, Singapore, October 2004.

[3] I. Guskov and A. Khodakovsky. Wavelet compression of parametrically coherent mesh sequences. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, August 2004.

[4] L. Ibarria and J. Rossignac. Dynapack: space-time compression of the 3D animations of triangle meshes with fixed connectivity. *ACM Symp. Computer Animation*, pages 126–135, 2003.

[5] Z. Karni and C. Gotsman. Compression of soft-body animation sequences. *Computers and Graphics*, 28:25–34, 2004.

[6] F. Payan and M. Antonini. 3D multiresolution context-based coding for geometry compression. In *Proceedings of IEEE International Conference in Image Processing (ICIP)*, Barcelona, Spain, sept. 2003.

[7] F. Payan and M. Antonini. An efficient bit allocation for compressing normal meshes with an error-driven quantization. *Computer Aided Geometric Design, Special Issue on Geometric Mesh Processing*, 22:466–486, July 2005.

[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] F. Payan and M. Antonini. Mean square error approximation for wavelet-based semiregular mesh compression. *IEEE Transactions on Visualization and Computer Graphics*, 12(4), July-August 2006.

[10] F. Payan, Y. Boulfani, and M. Antonini. Temporal lifting scheme for the compression of animated sequences of meshes. In *Proceedings of IEEE International Workshop VLBV 2005*, Sardinia, Italy, september 2005.

[11] C. Touma and C. Gotsman. Triangle mesh compression. *Graphics Interface’98*, pages 26–34, 1998.