Rate-Distortion Optimized Compression Algorithm for 3D Triangular Mesh Sequences

Meha Hachani, Azza Ouled-Zaïd, William Puech

Rate-Distortion Optimized Compression Algorithm For 3D Triangular Mesh Sequences

M. Hachani[†*], A. Ouled Zaid[†], and W. Puech[*]

[†]University of Tunis El Manar
Institut Suprieur d'Informatique
2 rue Abou Rayhan Bayrouni
2080 Ariana, Tunisia

[*]Montpellier University
LIRMM UMR CNRS 5506
161 rue Ada
34095 Montpellier Cedex 5, France

Abstract

This paper presents a new segmentation-based compression scheme for 3D dynamic models. The segmentation process is preformed based on the heat diffusion properties, while exploiting temporal and spatial dependencies of the geometry component. 3D affine transforms are used to describe the motion clusters. Weighting of these affine transforms allows to faithfully determine the vertex motions in each cluster. Indeed, both of affine transforms and weighting parameters are determined in order to estimate the positions of vertices in each frame, and consequently compute the temporal prediction errors to be quantized and encoded. In order to improve the coding efficiency, the prediction error quantization is optimized using a rate control mechanism. Comparative coding tests, for 3D mesh sequences, were conducted to evaluate the coding performance and examine the effectiveness of the proposed coder. The experiments have shown that our codec provides very satisfactory results when compared to the state of the art.

Introduction

In the last decade, the technological progress in the fields of telecommunication, hardware design and multimedia, allows access to an ever finer three-dimensional (3-D) modeling of the world. The critical challenges with 3D models lie in their visualization, rendering, protection or transmission over channels with limited bandwidth and storage on media with low capacity. While the majority of research in this area lies on 3D objects, now the trend is to turn to the 3D time domain (3D+t). 3D dynamic meshes are becoming a media of increasing importance. They constitute a fundamental and time consuming task in 3D animation systems. The 3D dynamic shapes are usually represented by a sequence of 3D meshes sharing the same connectivity and temporal information provided by time-varying geometry.

Research in the field of dynamic 3D mesh compression is focused on an analysis step that transforms the geometry information to reduce the signal amount of data, and then a differential coding strategy is applied on the resulting details to exploit the temporal dependencies between successive frames.

The pioneer compression algorithm designated for 3D dynamic meshes is proposed by Lengyel [8]. The main idea consists in partitioning a 3D mesh into different regions based on heuristic approach. For each region, a rigid-body motion is computed using 3D affine transforms to describe the vertex displacements in the same region. Later, Shamir and Pascucci [19] have extended Lengyel's coding approach by introducing a multi-resolution coding scheme for temporal deformation.

In the same context Mamou *et al.* [9] proposed a 3D mesh compression scheme, based on a skinning animation technique. The segmentation process is based on 3D affine transforms in order to obtain the frame-wise motion of each region by weighting the corresponding affine transforms. Motion compensation is then performed and Discrete Cosine Transform (DCT) is applied to encode the residual errors.

A relevant compression method, proposed by Alexa and Müller [17], performs the PCA of geometry covariance matrix to reduce spatial correlation. Karni and Gostman [6] have extended this technique by applying a second-order linear predictive coding (LPC) on PCA components, to exploit the temporal coherence. This algorithm has been shown to be efficient only for sequences with few global motion. In addition, its is penalized by its high time and memory complexity. More recently, Lee *et al.* [18] proposed to improve Alexa and Müller's compression algorithm by optimizing the number of key-frames. The drawback of these methods is that the number of key-frames may be quite high. Consequently, their effectiveness is diminished when applied to dense meshes with high number of vertices. To exploit temporal coherence, Payan and Antonini [12] proposed a wavelet-based method, using a temporal wavelet filtering. During the encoding step, the quantization of the wavelet coefficients is optimized based on a bit allocation procedure.

Recently, Bici *et al.* [1] developed a predictive coding approach based on weighted spatial prediction. The authors introduced an angle-based predictor during the refinement step. The proposed structure achieves a significant improvement in terms of prediction error accuracy and compression rate. More Recently, Vasa *et al.* [15] proposed a dynamic 3D mesh compression scheme, which exploits geometric laplacians. The main idea is to calculate only an average mesh for the whole sequence. The latter is used by applying a discrete geometric Laplacian, to encode the coefficients describing the mesh vertex trajectories. In order to improve the compression performance, the authors have integrated a spatio-temporel predictor.

The main purpose of this work is to propose a new coding method based on motion estimation of successive frames. Intrinsic mesh surface segmentation has been studied in the field of computer vision, especially for compression and simplification purposes. Therefore we use a segmentation-based compression scheme for animated 3D meshes with constant connectivity. The proposed segmentation method exploits the temporal coherence of the geometry information by using the heat diffusion properties. The motion of the resulting regions is accurately described by 3D affine transforms. These transforms are computed according to the first frame to match the subsequent ones. In order to improve the performance of our encoder, the quantization of the temporal prediction errors is optimized by using a bit allocation procedure. The objective aimed at is to optimize the quantization of the mesh geometry by minimizing the reconstruction error. We will experimentally show that compared to the reference methods, the proposed coding scheme offers good compression performance. Furthermore, the rate control mechanism, allows supporting specific needs of applications, networks and terminals.

The rest of this paper is organized as follows. In Section we describe in detail the various blocks constituting our coding scheme. In Section we investigate the performance of our compression system. Finally, in Section , we conclude with our final remarks and present some future works.

The Proposed Approach

This section presents the proposed compression algorithm destined to 3D dynamic meshes represented by a sequences of 3D triangular meshes sharing the same connectivity. The first mesh of the sequence is taken as a reference frame during the segmentation and the motion estimation. In what follows, we first describe the adopted segmentation approach, then, we introduce the overall compression algorithm illustrated in the block diagram of Fig. 1, and finally, we describe the rate control mechanism.

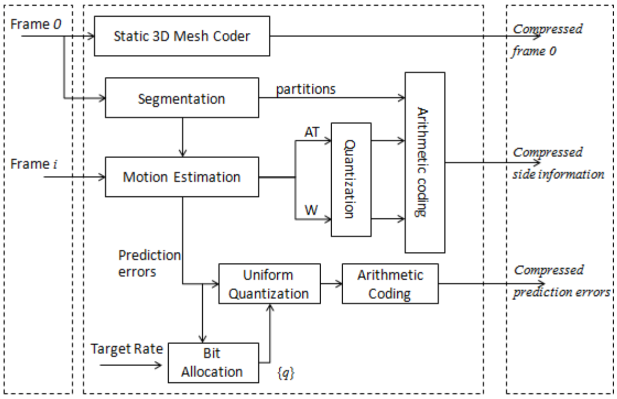


Figure 1: Block diagram of the proposed compression system.

Segmentation method

3D triangular surface meshes have a fairly wide descriptive power to manipulate in a simple way thanks to their algebraic simplicity and high usability. However this extrinsic representation suffers from high variability towards shape-preserving transformations like affine or isometric transformations. To overcome this problem it seems necessary to define computational intrinsic modeling. 3D mesh segmentation based on topological modeling, has emerged as an important issue thanks to the reduced cost of its implementation and the multitude of applications that are dependent on.

To efficiently estimate the motion information over time, we used an implicit segmentation algorithm presented in [4]. The main contribution consists in proposing a new scalar function μ that depends on some feature points, located on the extremity of the reference mesh. The latter are extracted based on the diffusion distance d_t^2 which allows to detect the most significant information of the object. The determined feature points F will be taken as the initial data to calculate the scalar function μ based on the eccentricity of the diffusion distance. For a practical implementation, the μ function is computed using a discrete formulation. For each vertex $v \in M$, $\mu(v)$

is approximated as the sum of the eccentricities from v to each feature point $p \in F$. As a result $\mu(v)$ is established as follows:

$$\mu(v) = \frac{1}{area(M)} \sum_{p \in F} d_t^2(v,p) area(p), \tag{1}$$

where $area(M)$ corresponds to the surface area of M, F the set of feature points which are extracted in the first step and $area(p)$ is the area of p neighborhoods. For more details on scalar function definition, the reader is referred to [5].

The obtained vector contains all the values of μ function is divided into intervals according to the value of μ and the number of connected components associated with each interval. In order to reduce the number of clusters, intervals with the same number of connected components and sharing a common border are fused into a single interval iteratively until obtaining intervals with different number of connected components.

Computing the scalar function μ in the discrete case may adversely affects the detection of region boundaries. For this reason, a refinement step is performed to adjust region boundaries with respect to deep surface concavities. Each region boundary is a level set and is thus associated with a value of μ that corresponds to a critical point. The optimal value of μ_{opt} should characterized a boundary that correspond to a concavity profile on the shape. It should be near to μ_c that match the closest critical point.

The objective is to search the optimal value μ_{opt} that corresponds to the boundary profile. This optimization problem consists to minimize the concavity function $E_{concave}(\mu\prime)$, given in eq. 2, of each region boundary associated with a value of $\mu\prime$.

$$E_{concave}(\mu\prime) = \min_{\mu\prime}(K_{min}(c_{(\mu\prime),R}(t)) \otimes G_\sigma(t)), \tag{2}$$

where $K_{min}(.)$ is a function which returns the sequence of K_{min} curvature values along the boundary profile, $c_{(\mu\prime)R}(t)$ is the curve-parameterized with respect to the normalized arc-length t. Indeed, $c_{(\mu\prime)R}(t)$ represents the portion of the μ values set corresponding to the boundary of region R.

Fig. 2 depicts some segmentation results for Dance and Snake sequences. We can see that the segmentation process allows partitioning the mesh into rigid clusters composed of topologically connected vertices which are characterized by similar motion properties.

Compression scheme

The proposed compression system represents the geometry of the mesh sequence by a piecewise affine geometry predictor that minimizes the prediction errors. After the segmentation process, a key-vector $\pi = \{1, .., V\}$ is established in order to associate each vertex to the index k of the cluster to which it belongs. This key-vector is directly encoded using lossless arithmetic coder. The connectivity is coded only once, together with the geometry information of the reference frame, using a static 3D mesh encoder. We privilege using the progressive lossless mesh encoder of Valette *et al.*

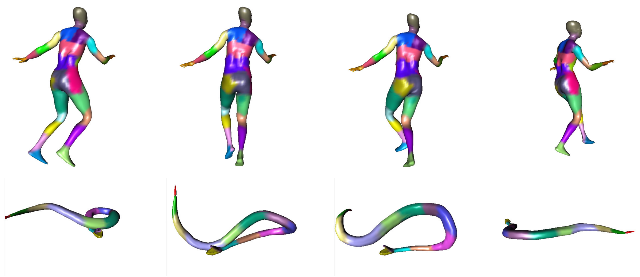


Figure 2: Segmentation results for selected frames extracted from Dance (a-b-c-d) and (e-f-g-h) Snake sequences.

[14], which is based on Incremental Parametric Refinement (IPR). In our setting, the vertices coordinates are quantized to 12 bits per coordinate.

The geometry of the remaining frames are encoded using a differential coding approach. Instead of coding the vertex coordinates in each frame, we take into account only their displacements over time. During the motion estimation stage, the affine motion of clusters $k \in \{1, .., K\}$ at frames $i \in \{1, .., F\}$ is described by matrix AT_i^k that represents the affine transforms. This matrix is computed as follows according to the reference frame:

$$AT_i^k = \underset{A}{\operatorname{argmin}}(\sum_{v \in \pi} \|A_{\chi_v^0} - \chi_1^v\|^2), \tag{3}$$

where A is 4×4 affine transform matrix, and χ_i^v is a 4D vector representing the homogeneous coordinates of the vertex v at frame i. The piecewise affine predictor of the frame i is expressed as follows:

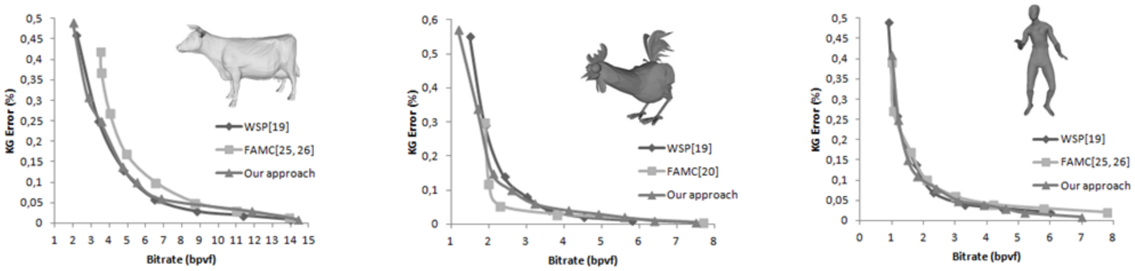
$$\hat{\chi}_i^v = \sum_{k=1}^{K} W_k^v AT_i^k \chi_0^v, \tag{4}$$

where W_k^v, defined in (4), is the optimal weight vector. It allows controlling the motion influence of all the clusters $k \in \{1, .., K\}$ over the vertex v. The motion of the vertices in each cluster is obtained by weighting the corresponding affine transforms. The estimated coordinates of the vertex v at frame i are computed from the set AT_i^k and W_k^v.

$$W_k^v = \underset{A}{\operatorname{argmin}} \sum_{i=0}^{F-1} \| \sum_{k=1}^{K} AT_i^k \chi_0^v - \chi_i^v \|. \tag{5}$$

Similarly to the vertices coordinates, the AT_i^k and W_k^v matrices are quantized to 12 bits per element. After that, they are encoded using the entropy arithmetic coder.

The prediction error is computed as a simple difference between the coordinates of the vertex v at frame i, and the estimated one. The obtained prediction error vector is represented by floating point components with low-magnitudes (that tend to zero). We proposed using the normalized scientific notation of the floating point components, to encode separately the signs, exponents and mantissas of the original



(a) (b) (c)

Figure 3: Rate-distortion performances for Cow (a), Chicken (b) and Dance (c) sequences.

vertex and the estimated one. This process allows realizing lossless compression with high precision.

It is important to note that the compressed file is composed of two kind of information:

- side information, which contains the key-vector $\pi = \{1, .., V\}$ and the two matrices AT_i^k and W_k^v,
- outside information, that contains the geometry and the connectivity of the reference frame as well as the prediction error vector.

bit allocation strategy

The main challenge of the proposed compression approach is to optimize the rate-distortion trade-off that offers a high compression performance while preserving the geometrical features of the input original geometry. The rate-distortion optimization is performed only on the prediction error vector, which is characterized by 3 subbands corresponding to the x, y, and z coordinates. The rate-distortion tradeoff is optimized by determining the best quantizer step size, for each subband, with respect to the target bit-rate R_{budget}. For a given subband S_i and a quantizer step size q, $D_i(q)$, the contribution to the total distortion from subband S_i, is defined by the mean square error, given by:

$$D_i(q) = \frac{1}{3} Mean\{(S_i - ((S_i//q) \times q))^2\}, \tag{6}$$

where the mean is calculated over all coordinates in S_i, and "// " denotes division followed by rounding to the nearest integer. Similarly, the total bit-rate contribution, $R_i(q)$, is defined by:

$$R_i(q) = \frac{1}{3} Entropy\{(S_i//q)\}, \tag{7}$$

where the entropy is calculated over all the coordinates in S_i using statistical models. The side information is encoded without loss, and consequently, their compression does not contribute to the induced distortion. Indeed, the total distortion is calculated by considering only the quantization error. The quality degradation is evaluated using the KG error introduced by Karni and Gotsman [6].

To calculate the total bit-rate, both side and outside information are considered. Given an input 3 subbands and a target bit-rate R_{budget}, one wants to select the set of quantization step sizes $Q = \{q_i : i = 1, \ldots, 3\}$ to minimize the total distortion $D(Q)$:

$$D(Q) = \sum_{i=1}^{3} D_i(q_i), \tag{8}$$

subject to the bit-rate constraint,

$$R(Q) = \sum_{i=1}^{3} R_i(q_i) \leq R_{budget} - R_s - R_{ff}. \tag{9}$$

Using Lagrange multiplier, equations 8 and 9 are equivalent to the following unconstrained problem:

$$\min_{Q} J(\lambda) = D(Q) + \lambda((R(Q) + R_s + R_{ff}) - R_{budget}), \tag{10}$$

where R_s and R_{ff} are the bit-rate of the side information and the reference frame, respectively. λ being the Lagrangian multiplier, and $J(\lambda)$ is the Lagrangian cost. For a fixed λ, $J(\lambda)$ is minimized when:

$$\frac{\partial J(\lambda)}{\partial D(Q)} = 0. \tag{11}$$

$$\frac{\partial J(\lambda)}{\partial R(Q)} = 0. \tag{12}$$

It is important to note that for a given Lagrange multiplier, the resulted λ and $J(\lambda)$ might not meet the overall rate constraint. Therefore, we should find the optimal Lagrange multiplier λ^{opt} obtained by using the bisection method [7].

Experimental Results

To evaluate the accuracy of the proposed segmentation method, we use the mean square error, introduced by the motion compensation procedure. The objective aimed at is to obtain a partition $\Pi = (\pi_k)_{k \in 1,..,K}$ of the whole mesh into K regions. The mean square error denoted by $E(\Pi)$ is defined by:

$$E(\Pi) = \frac{1}{V \times T \times D^2} \sum_{i=1}^{T} \sum_{k=1}^{K} \sum_{v \in \pi_k} \|\chi_i^v - A_i^k \chi_1^v\|^2,$$

where V and T denote the number of vertices and the number of frames of the mesh sequences, respectively. D is the bounding box diagonal of the first frame. A_i^k is the 3D affine transform associated with the region (π_k) at frame i, and χ_i^v is a vector that contains the homogeneous coordinates of the vertex v at frame i.

Table 1 lists the values of the squared error $E(.)$ before and after the refinement step. The number of iterations has also been provided in order to evaluate the convergence rate. Based on the results reported in Table 1 we can notice that, on average,

Table 1: Evaluation of the mean square error of the motion compensation $E(\Pi)$.

	$E(\Pi_0)$	Nb iterations	$E(\Pi)$
dance	0.0023	19	0.0012
Cow	0.0028	16	0.0016
Snake	0.0024	25	0.0014

the refinement step converges after 20 iterations. The motion estimation error induced when integrating the refinement step is about 0.0014 against 0.0025 without. Thus we conclude that, the refinement post-processing stage allows to increase, significantly, the motion estimation accuracy.

To perform objective comparisons, the simulations were conducted on three 3D dynamic meshes: Dance, Chicken, and Cow. These models are characterized by their various motions. They offer a good variability in terms of spatial and temporal sizes. Table 2 summarizes their properties, expressed in terms of numbers of vertices, number of frames, and number of connected components. To assess the efficiency of our compression scheme, we compare its effectiveness to those of WSP [1] and FAMC [10, 13] algorithms. The Weighted spatial prediction (WSP) [1] algorithm integrates three prediction structures (weighted spatial prediction with its weighted refinement and angular based predictor). Frame-based Animated Mesh Compression (MPEG-4 FAMC) [10, 13] is based on a skinning approach [9], coupled with context-adaptive binary arithmetic coding.

Table 2: Properties of the tested dynamic meshes.

	Vertices	Frames	Connected Components
Cow	2904	204	1
Dance	7061	201	1
Chicken	3030	400	41

Fig. 3 depicts the rate/distortion curves obtained when using our codec and the reference algorithms for Cow, Chicken, and Dance models. From this figure we can see that, for all the tested bit-rates, our coding scheme yields superior compression performance when compared to WSP and FAMC. Furthermore, the rate/distortion curves show that FAMC offers the worst results, except for Chicken sequence, which seems to be better compressed by FAMC.

To reduce the motion estimation error, our coder uses a faithful segmentation technique based on heat diffusion properties. Also the accuracy of the vertices distribution, on the border between clusters, has been enhanced by exploiting the curvature information. All this reasons justify the good results obtained in terms of distortion when comparing our method to the reference one.

Finally, we measure the root mean square error (RMSE) individually for each frame obtained at $7bpvf$ and $5bpvf$ for Cow and Dance models, respectively. The number of clusters, K, was fixed to 30. Examining Fig. 4(a) and (b) showing the

RMSE curves, we notice that our method produces low compression distortions. The reported values of RMSE (10^{-5}) vary in the range of $[6.8, 9.4]$ and $[5.7, 11.1]$ for the Cow and Dance sequences, respectively.

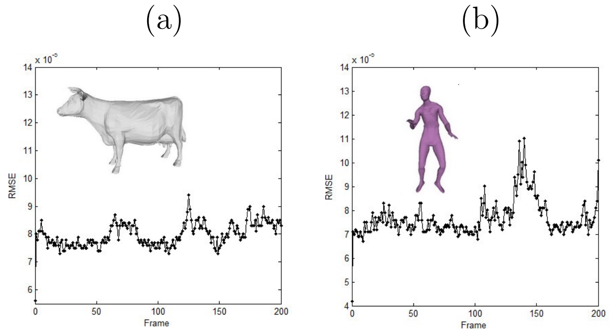


Figure 4: Distortion curves for Cow and Dance sequences.

Conclusion

In this paper, we proposed a segmentation based compression system adapted to animated 3D meshes. The segmentation approach allows partitioning the mesh vertices into clusters based on the heat diffusion properties. The obtained partition vector is used to compute a piecewise affine predictor which minimizes the prediction errors. The rate-distortion efficiency of the proposed encoder is improved by optimizing the quantization of the temporal prediction errors using a rate control mechanism. Experimental results have shown that our approach leads to satisfactory results compared to the state of the art.

In our future work, we plan to improve the performance of the proposed scheme in terms of reconstruction quality. The main idea consists in automatically select the number of clusters that provides a lower quality degradation.

References

[1] M. O. Bici and G. B. Akari, "Improved prediction methods for scalable predictive animated mesh compression ", J. Vis. Commun. Image R., vol. 22, pp. 577–589, 2011.

[2] J.W. Cho, S. Valette, J. H. Park, H. Y. Jung, and R. Prost, "3-D mesh sequence compression using wavelet-based multi-resolution analysis ", Applied Mathematics and Computation, vol. 216, pp. 410–425, 2010.

[3] I. Guskov and A. Khodakovsky, "Wavelet compression of parametrically coherent mesh sequences ", in Eurographics/ACM SIGGRAPH symposium on computer animation, 2004.

[4] M. Hachani, A. Ouled Zaid, W. Puech, "Kinematic Reeb Graph Extraction Based on Heat Diffusion ", 22nd IEEE International Conference on Pattern Recognition, 2014, pp. 3981–3986.

[5] M. Hachani, A. Ouled Zaid, W. Puech, "Segmentation of 3D Dynamic Meshes Based on Reeb Graph Approach. ", 22nd European Signal Processing Conference, 2014, pp. 2175–2179.

[6] Z. Karni, C. Gostman, "Compression of soft-body animation sequences ", Computer and Graphics, vol. 28, pp. 25–34, 2004.

[7] B. S. Krongold, and K. Ramchandran, and D. L. Jones, "Computationally efficient optimal power allocation algorithm for multicarrier communication systems ", Proceedings of International Conference on Communications, 1998. pp. 1018–1022.

[8] J. Lengyel, "Compression of time-dependent geometry ", ACM Symposium on Interactive 3D Graphics, 1999, pp. 89–96.

[9] K. Mamou, T. Zaharia, and F. Prteux, "Multi-chart geometry video: A compact representation for 3D animations ", IEEE 3DPVT, 2006, pp. 711–718

[10] K. Mamou, T. Zaharia, F. Preteux, "FAMC: The mpeg-4 standard for animated mesh compression ", 15th IEEE International Conference on Image Processing, 2008, pp. 2676-2679.

[11] 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, 2005. pp. 466–486.

[12] F. Payan and M. Antonini, "Temporal wavelet-based compression for 3D animated models ", Computers and Graphics, vol. 31, pp. 77–88, 2007.

[13] N. Stefanoski, J. Ostermann, "Spatially and temporally scalable compression of animated 3D meshes with MPEG-4/FAMC ", in: Proceedings of the IEEE International Conference on Image Processing, pp. 2696–2699, 2008.

[14] S. Valette, R. Chaine, and R. Prost, "Progressive Lossless Mesh Compression Via Incremental Parametric Refinement ", Eurographics Symposium on Geometry Processing, vol. 28, no. 5, pp. 1301–1310, 2009.

[15] L. váša, S. Marras, K. Hormann, G. Burnett, "Compression Dynamic Meshes with Geometric Laplacians ", Computer and Graphics Forum, vol. 33, no. 2, pp. 145–154, 2014.

[16] K. Mueller, A. Smolic, M. Kautzner, P. Eisert, "Rate-distortion-optimized predictive compression of dynamic 3d mesh ", SP:IC, vol. 9, pp. 812–828, 2006.

[17] M. Alexa, W. Müller, "Representing animations by principal components ", Computer and Graphics Forum. vol. 19, no. 3, pp. 411–418, 2000.

[18] P.F. Lee, C.K. Kao, J.L. Tseng, B.S. Jong, T.W. Lin , "3D animation compression using affine transformation matrix and principal component analysis ", IEICE - Trans. Inf. Syst. E90-D, vol.7, pp. 1073–1084. 2007.

[19] A. Shamir and V. Pascucci, "Temporal and spatial level of details for dynamic meshes ", In: Proceedings of virtual reality systems and techniques, 2001. pp. 423–430.