Rev. Roum. Sci. Techn.– Électrotechn. et Énerg.
Vol. **61**, *2*, pp. 201–204, Bucarest, 2016

Automatique et ordinateurs

GEOMETRY COMPRESSION FOR 3D MODELS

MIRCEA CURILĂ, SORIN CURILĂ, CRISTIAN GRAVA[1]

Key words: Geometry compression, 3D models, Vector quantization, Blind source separation (BSS), Arithmetic coding.

The 3D models, represented by vertex positions and the association between triangles and vertices, colors, normals and texture information, are stored as virtual reality modeling language (VRML) format files. Connectivity information and photometry information are not treated in this paper. The vertices coordinates follow a vector quantization process in order to obey the bitrate control constraint. Next, a blind source separation algorithm is proposed to perform a decorrelation phase. At the end, arithmetic coding is applied to form the final bit steam, thus containing the compressed 3D information. The obtained experimental results are performant, being comparable with those of actual compression methods of 3D models, thereby proving an actual gain. Thus, the technique based on the BSS algorithm offers an alternative way for the compression of 3D models.

1. INTRODUCTION

The principle of geometry encoding requires the following steps:

a) the coordinates are uniformly quantized, the quantization step being chosen by an iterative search algorithm to control the number of bits per vertex [1];

b) data decorrelation is performed by a Blind Source Separation method, which takes the correlated geometry as observations and decorrelated geometry as sources;

c) the decorrelated components and mixing matrix are encoded using the arithmetic coding.

Blind source separation (BSS) is a tool for estimating original sources from their mixtures at multiple sensors [2, 3]. More natural sources generate the signals $\boldsymbol{S}$, and after their propagation signals $\boldsymbol{X}$ are received on multiple sensors. The source signals $\boldsymbol{S}$ can be considered as a stationary multivariate process, mutually uncorrelated.

Next it will be shown how 3D model geometry can be expressed as a linear combination of decorrelated geometry's components making possible to apply BSS algorithm for data decorrelation.

2. VECTOR QUANTIZATION

This algorithm assumes a lossy data compression. Thus, for a vector $\boldsymbol{X} = \{X_1, X_2, \ldots, X_N\}$, having mean square error (MSE) as distortion measure and $\boldsymbol{Nc}$ number of codevectors, the algorithm finds a $\boldsymbol{C}$ codebook and a $\boldsymbol{P}$ partition in the smallest average distortion.

The method uses a training sequence (***TS***) obtained from 3D data of several known models:

$$\begin{aligned} &\mathbf{TS} = \{X_1, X_2, \ldots, X_{\dim}\} \\ &\boldsymbol{X_k} = \left(X_{kx}, X_{ky}, X_{kz}\right), k = 1,\ 2,\ \ldots,\ \dim \end{aligned} . \quad (1)$$

$\boldsymbol{C}$ denotes the codebook, $\boldsymbol{C}=\{c_1, c_2, \ldots, c_{Nc}\}$, where $c_k = (c_{kx}, c_{ky}, c_{kz})$ is a 3D message, $k = 1, 2, \ldots, Nc$.

Considering $\boldsymbol{P_k}$ the encoding volume associated at c_k message and $\boldsymbol{P} = \{P_1, P_2, \ldots, P_{Nc}\}$ a partition of the entire volume, if vector X_p belongs to the encoding volume $\boldsymbol{P_q}$, its approximation q_{X_p} is done by: $q_{X_p} = c_q$. The representative message c_q is determined to be the closest in average distortion from the X_p vector:

$$\mathbf{MSE} = \frac{1}{3\ \dim} \sum_{p=1}^{\dim} \left\| X_p - c_q \right\|^2 . \quad (2)$$

The codebook design problem finds $\boldsymbol{C}$ and $\boldsymbol{P}$ such that MSE is minimized. There is an initial codebook $\boldsymbol{C}^{(0)}$, where the initial message is set as the average of the entire TS. Next, the message is split into two and the algorithm runs with the two messages as the initial codebook. The splitting process continues until the value of fixed Nc is achieved. The description of the algorithm is presented in the following lines:

- for a given TS and ε;
- set $c_1^* = \frac{1}{\dim}\sum_{p=1}^{\dim} X_p$ and Nc = 1. Compute

average distortion: $\mathrm{MSE}^* = \frac{1}{3\dim}\sum_{p=1}^{\dim}\left\|X_p - c_1^*\right\|^2$;

- for (i = 1; i <= Nc; i++) {
  $c_i^{(0)} = (1+\varepsilon)\, c_i^*$; $c_{Nc+i}^{(0)} = (1-\varepsilon)\, c_i^*$; }
  $Nc = 2{*}Nc$;
- set $i = 0$ and $\mathrm{MSE}^{(0)} = \mathrm{MSE}^*$

1. for (p = 1; p <= dim; p++) {
   Find the minimum value of squared-error distortion $\left\|X_p - c_q^{(i)}\right\|^2$, q = 1, 2, …, Nc. Calculate approximation q_{X_p} for the index that achieves the minimum (denoted q^*) , as: $q_{X_p} = c_{q^*}^{(i)}$; }
2. for (q = 1; q <= Nc; q++) {
   Compute the new message:
   $$c_q^{(i+1)} = \frac{\sum_{q_{X_p} = c_{q^*}^{(i)}} X_p}{\sum_{q_{X_p} = c_{q^*}^{(i)}} 1} \ ; \}$$
3. increment index i: i++;
4. compute average distortion:
   $$\mathrm{MSE}^{(i)} = \frac{1}{3\dim}\sum_{p=1}^{\dim}\left\|X_p - q_{X_p}\right\|^2 ;$$

[1] University of Oradea, Universității street, no. 1, 410087 – Oradea, Romania, E-mail: mirceacurila@yahoo.com, scurila@uoradea.ro, cgrava@uoradea.ro

5. if $\dfrac{\left(\mathrm{MSE}^{(i-1)} - \mathrm{MSE}^{(i)}\right)}{\mathrm{MSE}^{(i-1)}} > \varepsilon$ go to 1.

6. $\mathrm{MSE}^{*} = \mathrm{MSE}^{(i)}$;
for ($q = 1$; $q <= Nc$; q++)
$c_q^{*} = c_q^{(i)}$.

Encoding process takes a vector X_p and outputs the message c_q that offers the lowest distortion in the codebook $\boldsymbol{C}$.

3. DECORRELATION BY BLIND SOURCE SEPARATION

Some methods in blind source separation (BSS) and independent components analysis (ICA) use higher order statistics to obtain statistically independent sources [4]. Other approaches such as Karhunen-Loève transform (KLT) [5], eigenvalues decomposition (EVD) [6] and second order blind identification (SOBI) [7] exploit the temporal, spatial or spectral diversities of the sources. This is achieved by using delayed covariance matrices at different delays to impose a decorrelated structure on the solution.

When the input vector elements are highly correlated, like the geometry of 3D model, by its processing with a BSS algorithm, the transformed vector elements tend to be uncorrelated. The dimension of the transformed vector can be reduced by neglecting eigenvectors that correspond to small eigenvalues. Thus, only the coefficients corresponding to the highest energy are retained. The inverse transform gives a reconstruction of the original vector with loss.

Let X_i, $1 \leq i \leq N$ be a vertices sequence quantified by qX_i values. With a linear prediction rule, it is possible to estimate $q\hat{X}_i$. The difference between the value of the current vertex qX_i and its estimate $q\hat{X}_i$ is called "prediction error" $\mathrm{d}X_i$. Only these errors are coded in predictive techniques. This rule was first proposed by C. J. Kuo [8] and uses the coefficients α, β and γ to predict the current vertex X from the three already crossed vertices belonging to the same polygon:

$$X = \alpha X_i + \beta X_j + \gamma X_k, \text{ with } \alpha + \beta + \gamma = 1. \quad (3)$$

Using the predictive method [8] it is possible to express the geometry vector $\boldsymbol{g}$ as follows:

$$\begin{cases} X_1(x,y,z) = \mathrm{d}X_1 \\ X_2(x,y,z) = A_{2,1}\,\mathrm{d}X_1 + \mathrm{d}X_2 \\ X_3(x,y,z) = A_{3,1}\,\mathrm{d}X_1 + A_{3,2}\,\mathrm{d}X_2 + \mathrm{d}X_3 \\ \cdots\cdots \\ X_s(x,y,z) = A_{s,1}\,\mathrm{d}X_1 + A_{s,2}\,\mathrm{d}X_2 + \ldots + A_{s,s-1}\,\mathrm{d}X_{s-1} + \mathrm{d}X_s, \end{cases} \quad s \in \{4,\ldots,N\} \quad (4)$$

The matrix of observations denoted $\boldsymbol{X}$ is associated to the initial geometry, its columns containing N values of the vertices for each direction of the Cartesian space. The sources' matrix $\boldsymbol{S}$ corresponds to the decorrelated geometry coded by predictive compression method:

$$\boldsymbol{S} = \begin{pmatrix} s_1(1) & s_1(2) & s_1(3) \\ s_2(1) & s_2(2) & s_2(3) \\ \ldots & \ldots & \ldots \\ s_M(1) & s_M(2) & s_M(3) \end{pmatrix} = \begin{pmatrix} \mathrm{d}X_1(x) & \mathrm{d}X_1(y) & \mathrm{d}X_1(z) \\ \mathrm{d}X_2(x) & \mathrm{d}X_2(y) & \mathrm{d}X_2(z) \\ \ldots & \ldots & \ldots \\ \mathrm{d}X_M(x) & \mathrm{d}X_M(y) & \mathrm{d}X_M(z) \end{pmatrix}. \quad (5)$$

Thus, the initial geometry $\boldsymbol{g}$ is expressed as a linear combination of the components of the decorrelated geometry $\mathbf{d}\boldsymbol{g}$.

The geometry description of the 3D model in the VRML file shows a strong spatial correlation in each direction of the Cartesian coordinate system [9,10]. It is possible to decorrelate the data by BSS methods. In this context, the initial geometry vector $\boldsymbol{g}^{\mathrm{T}} = [X_1, X_2, \ldots, X_N]^{\mathrm{T}}$ is expressed by a vector $\mathbf{d}\boldsymbol{g}^{\mathrm{T}}$ with M decorrelated components $\mathbf{d}\boldsymbol{g}^{\mathrm{T}} = [\mathrm{d}X_1, \mathrm{d}X_2, \ldots, \mathrm{d}X_M]^{\mathrm{T}}$, associated with a mixing matrix $\boldsymbol{A}[N\times M]$, where $M << N$.

Because the number of extracted sources is very small, their values are included in the header of the compressed file. Mixing matrix elements being uncorrelated, determine that their corresponding binary data contain more identification bits and less refinement bits. Matrix elements are quantified, binarized through successive approximation and coded using the arithmetic code, which strongly compresses the binary information consisting in more consecutive bits of the same value 0 or 1 suitable to the identification bits.

At reception 3D model geometry is restored by mixing the decorrelated geometry (the extracted sources) using the arithmetic decoded and dequantized mixing matrix.

The M columns of the mixing matrix $\boldsymbol{A}$ have the same dimension (N) as the geometry of the 3D model. To reduce this disadvantage that affects the compression, the geometry is divided into b blocks, denoted $\boldsymbol{bg}$, of dimension N/b:

$$\boldsymbol{bg}(i) = \boldsymbol{g}\left((i-1)\frac{N}{b} + 1 : i\frac{N}{b}\right), i = 1, \ldots, b. \quad (6)$$

Each block provides a covariance matrix $\boldsymbol{R}_{\boldsymbol{X}_i}(0)$. These matrices are averaged to give a global covariance matrix. The diagonalization of this matrix provides a whitening matrix $\boldsymbol{W}$ which is then applied to each block of the geometry:

$$\underline{\boldsymbol{bg}}[M\times 3] = \boldsymbol{W}\left[M \times \frac{N}{b}\right]\cdot \boldsymbol{bg}\left[\frac{N}{b}\times 3\right]. \quad (7)$$

For each block of whitened geometry a delayed covariance matrix $\boldsymbol{R}_{\boldsymbol{Y}_i}(p),\ p \neq 0$ is determined. The averaging of these matrices provides a global delayed covariance matrix. The diagonalization of this matrix leads to a global unitary matrix $\boldsymbol{U}$, used to separate the sources in Cichocki's algorithm [6]. By applying the transposed of the global unitary matrix $\boldsymbol{U}$ to each block of the whitened geometry $\underline{\boldsymbol{bg}}$ the decorrelated geometry blocks are obtained:

$$\boldsymbol{b}\mathrm{d}\boldsymbol{g}[M\times 3] = \boldsymbol{U}^{\mathrm{T}}[M\times M]\cdot \underline{\boldsymbol{bg}}[M\times 3]. \quad (8)$$

Consequently, the dimension of the mixing matrix is decreased from $N\times M$ to $N/b\times M$, and that of the decorrelated geometry is increased from M to bM. The number of blocks must be chosen so that it should lead to a tradeoff between the size of mixing matrix and that of the decorrelated geometry, fact that can ensure the decrease of the mixing matrix size as much as possible, altogether with an increase of the decorrelated geometry size as little as possible, without introducing significant errors at the reconstruction of the models.

4. ARITHMETIC CODING

Arithmetic coding stops the representation of each symbol of the input source through a specific code and, together with this, stops the limitation to assign to each symbol an integer number of bits. Thus, it becomes possible to obtain an average length of code words, close to the source entropy.

Here, the messages are ordered on a disk of probability between [0, 1] for a known sequence to both encoder and decoder. To each message is assigned a subinterval equal to its probability. As the message becomes longer, the interval needed to highlight it becomes smaller, and the number of bits needed to mention that interval becomes higher. Successive messages reduce the size of the field in relation to the message probabilities. At the beginning of the transmission, the field of the messages is the interval [0, 1]. As each message is processed, the field is restricted to that portion allocated to the message. The decoder can see which subinterval is pointed to by the code stream and decode the appropriate message.

In the following, we will use the transformation $x(c_k) = k$; $c_k \in C$; $k = 1, 2, \dots, N_c$, where $C = \{c_1, c_2, \dots, c_{Nc}\}$ is the codebook of a discrete source and x is a random variable that can take the following values $1, 2, \dots, N_c$. For a given distribution of the source, a cumulative distribution function $f_x(k) = \sum_{i=1}^{k} P(x=i)$ can be associated at the random variable x. The distribution function is used to divide the interval of length 1 in subintervals $[f_x(k-1), f_x(k)]$ proportional to the values of this function. The emergence of first message in the sequence restricts the interval containing the label message to one of these subintervals. Next, this subinterval is partitioned into subintervals proportional to the values of the distribution function, just like the interval [0, 1].

Let $C = \{c_1, c_2, c_3, c_4\}$ be the source codebook, with the following distribution $P(c_1) = 0.5$, $P(c_2) = 0.25$, $P(c_3) = 0.125$, $P(c_4) = 0.125$. Using the relation $f_x(k)$ defined above, it results $f_x(1) = 0.5$, $f_x(2) = 0.75$, $f_x(3) = 0.875$ and $f_x(4) = 1$.

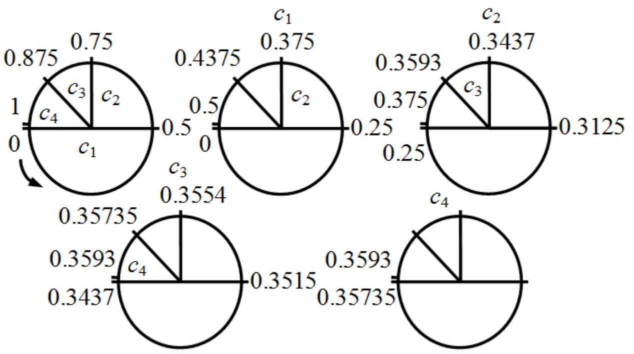

Fig. 1 – Intuitive representation of the ranges and their corresponding labels.

Initially, both the encoder and decoder know the first range (Figure 1). On the first disk of the figure, there were indicated the values 0.5, 0.75 and 0.875. The subinterval containing the label depends on the first message of the sequence to be encoded. If that is c_1, the label is in the range [0, 0.5]. For this example, the subinterval [0, 0.5], corresponding to the first message, is divided in the same proportions (0.5, 0.75 and 0.875) just like interval [0, 1]. On the second disk in Fig. 1, there are indicated the values 0.25, 0.375 and 0.4375 obtained in this way. Since the second message is c_2, the interval of length 0.125 is retained within the limits 0.25 and 0.375 and is divided proportionally to the numbers 0.5, 0.75, and 0.875. Thus, the values 0.3125, 0.3437 and 0.3593 indicated on the third disk are obtained. As the third message is c_3, the interval of length 0.0156 is retained, within the limits 0.3437 and 0.3593, which builds the fourth disk. The values 0.3515, 0.3554 and 0.35735 are indicated on this. As the fourth message is c_4, the interval within the limits 0.35735 and 0.3593 is still retained.

Proceeding in this way, the encoded message builds up as follows: initially [0, 1]; after computation: c_1 [0, 0.5], c_2 [0.25, 0.375], c_3 [0.3437, 0.3593], c_4 [0.35735 , 0.3593].

It isn't necessary for the decoder to know both ends of the range produced by the encoder. A single number within the range will suffice. In this paper we chose the labels as midpoints of intervals, using the relation:

$$L_x(c_k) = \sum_{i=1}^{k-1} P(x=i) + \frac{1}{2} P(x=k)\,,\ k = 1, 2, \dots, N_c \,. \quad (9)$$

Here, the *integer code* of $L_x(c)$ is obtained considering the binary representation of the number and truncating at $[\log_2 1/P(c)] + 1$, where $[\alpha]$ represents the lowest integer greater than or equal to α.

Thus, for the sequence c_1, c_2, c_3, c_4 it results $L_x(c_1)$: $0.25 = = 0*2^{-1}+1*2^{-2}$ and c_1 integer code: 01; $L_x(c_2)$: $0.625 = 1*2^{-1}+0*2^{-2}+1*2^{-3}$ and c_2 integer code: 101; $L_x(c_3)$: $0.8125 = 1*2^{-1}+1*2^{-2}+0*2^{-3}+1*2^{-4}$ and c_3 integer code: 1101; $L_x(c_4)$: $0.9375= 1*2^{-1}+1*2^{-2}+1*2^{-3}+1*2^{-4}$ and c_4 integer code: 1111.

5. EXPERIMENTAL RESULTS

The encoding method takes as inputs the VRML file and the bitrate (the required number of bits for encoding each vertex of the mesh), and generates a compressed file as output. The decoder takes the compressed file as input and produces a VRML file.

We have tested the proposed algorithm for three bitrates on several 3D models, including Beethoven, Carnation and Satellite (representative models, selected in this paper to present our results). The size of the fifty tested 3D models increases from a few faces to hundreds of thousands of faces.

The results were evaluated both subjectively and objectively. Subjective evaluation is done by visual inspection of the reconstructed models using a VRML browser. For objective evaluation, Frank Bossen developed a software for computing the distortion measure [11]. The used measure is defined as a metric between two 3D models with the same connectivity. For each vertex of the first model, the software finds its nearest vertex in the second one and reports the distance between them. In reverse order, for each vertex of the second model, the software finds its closest vertex in the first model and reports the distance between them. The mean value of these distortions for all vertices gives the e_r , that is the distortion measure.

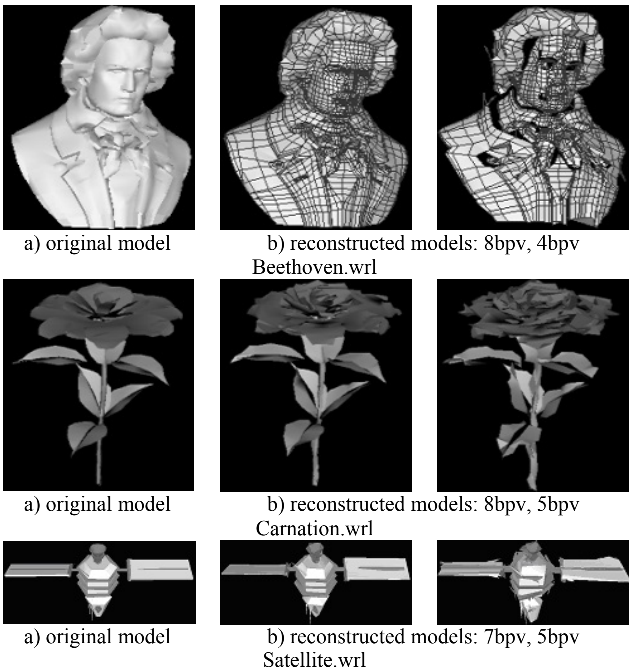

Fig. 2 – Subjective evaluation.

Experimental results on the 3D models named Beethoven, Carnation and Satellite are shown in Figure 2. The geometry reconstruction has been affected by both the quantization error and the error corresponding to neglected coefficients with small energy in the decorrelation process using BSS algorithm. The first image represents the original 3D model. The second image corresponds to a good reconstruction for which the bitrate and the number of sources are optimal and the reconstruction error is small. In the third image one can see the degradation by using a too small bitrate and number of sources.

The geometry compression performances for EVD algorithm [6], appreciated on both the geometry compression rate r[%] (the ratio between compressed geometry and original geometry of 3D model) and reconstruction error corresponding to some values of bitrate for three sources ($M = 3$) and two blocks ($b = 2$), are shown in Table 1. Each line of the table corresponds approximately to the same reconstruction error (with 10^{-1} tolerance, invisible to the human eye). The obtained results are performant, comparable with those of actual compression methods of 3D models.

Table 1

Objective evaluation

a) Beethoven.wrl		b) Carnation.wrl		c) Satellite.wrl	
bitrate	*r*[%]	bitrate	*r*[%]	bitrate	*r*[%]
8	85	8	87	7	85
6	86	7	88	6	89
4	89	5	93	5	91

6. **CONCLUSION**

Here, we addressed the issue of geometry compression of 3D models. Within the reconstruction of 3D models framework and under a bitrate control constraint, we studied the vector quantization, arithmetic coding and blind source separation algorithms for geometry compression.

The compression technique based on BSS algorithm decorrelates the initial geometry, preserving the information needed for reconstruction in the reduced mixing matrix and in the extracted sources. After processing the data number has not decreased. On the contrary, in addition to the matrix elements the values of the extracted sources have appeared. The gain obtained after processing using BSS method in terms of compression is the passing from the spatially correlated geometry data to the mixing matrix data that no longer have the same property. In contrast to the correlated geometry situation, the resulted binary data (having more identification bits and less refinement bits) allow to the arithmetic encoder an obvious superior compression to the case of the spatially correlated data.

The best results are obtained when the correlated geometry is divided into two blocks ($b = 2$), one separated three sources ($M = 3$) from each block of correlated geometry, and one used a global mixing matrix with dimension equals $N / 2 \times 3$.

The reconstructed models in Fig. 2 show the geometry changes as a function of the required number of bits per vertex. The simulation carried out on fifty 3D models, established that for a correct coded model, geometry compression rate is higher than 80 % at a given bitrate, which is an effective gain in 3D models compression.

The method of compression proposed in this paper has performances comparable with the actual methods and a detailed study in this regard will be presented in a further paper.

Received on November 19, 2015

REFERENCES

1. M. Deering, *Geometric compression*, ACM Computer Graphics Proceedings, pp. 13–20, SIGGRAPH ’95, Los Angeles, 1995.
2. M.N. Syed, P.G. Georgiev, P.M. Pardalos, *A hierarchical approach for sparse source Blind Signal Separation problem*, Computers & Operations Research, **41**, pp. 386–398, 2014.
3. Danielle Nuzillard, J.-M. Nuzillard, *Second-order blind source separation in the Fourier space of data*, Signal Processing, **83**, *3*, pp. 627–631, 2003.
4. A. Hyvarinen, *Fast and robust fixed-point algorithms for independent component analysis*, IEEE Trans. On Neural Networks, **10**, *3*, pp. 626–634, 1999.
5. K.R. Rao, P.C. Yip, *The Transform and Data Compression Handbook*, Boca Raton, CRC Press LLC, 2001.
6. A. Cichocki, S. Amari, *Adaptive Blind Signal and Image Processing. Learning Algorithms and Applications*, John Wiley and Sons, Ltd Baffins Lane, Chichester West Sussex, PO19 1UD, 2002.
7. A. Belouchrani, K. Abdel-Meraim, J.F. Cardoso, E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Processing, **45**, pp. 434–444, 1997.
8. J. Li, C.J. Kuo, *Embedded Coding of Mesh Geometry*, Research Report ISO/IEC JTC1/SC29/WG11, MPEG98/M3325, Tokyo, Japan, March, 1998.
9. G. Taubin, J. Rossignac, *Geometric compression through topological surgery*, Research Report, IBM Research, RC-20340(#89924), January 1996.
10. N.A. Rumman, S.A. El-Seoud, K.F. Khatatneh, C. Gütl, *Geometry Compression for 3D Polygonal Models using a Neural Network*, International Journal of Computer Applications (0975–8887), **1**, *29*, pp.13–22, 2010.
11. F. Bossen, *Description of Core Experiments for 3DMC*, Research Report ISO/IEC JTC1/SC29/m4061, Atlantic City, USA, October, 1998.
12. R.D. Albu, I. Dzitac, F. Popentiu-Vladicescu, I.M. Naghiu, *Input Projection Algorithms Influence in Prediction and Optimization of QoS Accuracy*, Int. J. Comput. Commun., **9**, *2*, pp. 131–138, 2014.