Geometry Compression of 3D Mesh Utilising Robust Second Order Blind Identification Algorithm

Mircea Curilă,

University of Oradea, Department of Physics-Informatics-Chemistry

str. Universităţii nr. 1, zip code 410087 Oradea, România

E-mail: mcurila@uoradea.ro

Sorin Curilă

University of Oradea, Department of Electronics

str. Universităţii nr. 1, zip code 410087 Oradea, România

E-mail: scurila@uoradea.ro

Abstract: The 3D mesh geometry is spatially correlated on each direction in the Cartesian coordinate system, this redundancy leading to a huge VRML file. In order to decorrelate the geometry of 3D mesh, the linear prediction rule used by actual methods is substituted by a *Blind Sources Separation* technique. For this reason we propose to take the correlated geometry of 3D mesh as observations and decorrelated geometry as sources corresponding at largest energy. The global mixing matrix, achieved by the geometry division into blocks, represents the bitstream of compressed file. Its elements are quantized, binary approximated and encoded using arithmetic coding, providing effective compression. *BSS* technique proposed is entitled *Robust Second Order Blind Identification*.

Keywords: Robust Second Order Blind Identification, eigenvalues decomposition, Prediction, 3D mesh, compression.

Mircea Curilă is Associate Professor in Informatics, Head of Physics-Informatics-Chemistry Department, Faculty of Environmental Protection, University of Oradea, Romania. His fields of interest are in image enhancement, higher order statistics, image compression, blind sources separation, virtual reality and virtual environment

Sorin Curilă is Associate Professor in Electronics Department, Faculty of Electrical Engineering and Information Technology, University of Oradea, Romania. His fields of interest are in image representation, image enhancement, higher order statistics and image compression. He has published more than 55 technical papers in International, National journals and conferences

1. Introduction

The Virtual Reality Modeling Language (VRML) has become the most commonly used standard for representing such 3D models. A VRML file contains complex information in text format related either to the connectivity and the geometry of the model, or to the model properties. Basically, a 3D mesh is defined by a set of vertices and a set of faces. The vertex location is defined by its coordinates in the 3D Cartesian system. A face is defined as an ordered sequence of vertex indices. The connectivity represents the relationships between vertices, the geometry refers to the position of the vertices and the properties contain photometric information, including color, texture and normals.

The VRML files needs effective 3D compression techniques that would significantly reduce the transmission time, the used memory and local disk space. A very big community of researchers has tried to find different algorithms to manipulate the 3D data. One mentions here those whose results are remarkable and very close to aim of our work: J. Rossignak [17], G. Taubin [18] [20], M. Deering [10], F. Bossen [13], F. Lazarus [18], M. Chow[5], C. Gostman [22], C. J. Kuo [14], F. Preteux [11], D. Nuzillard [7], A. Gueziec [12], etc. The compression procedure involves three different coding steps for the connectivity, the geometry and the properties of the mesh. Our contribution is a method concerning the geometry coding.

2. Robust Second Order Blind Identification (RSOBI)

In many practical problems the processed data are multidimensional observations, that has the form:

$$\boldsymbol{x}(k) = \boldsymbol{A}\ \boldsymbol{s}(k) \tag{1}$$

where the N-dimensional vector $\boldsymbol{x}(k) = [x_1(k), x_2(k),\ldots, x_N(k)]^T$ is an instantaneous linear mixture of source signals, the M-dimensional vector $\boldsymbol{s}(k) = [s_1(k), s_2(k), \ldots , s_M(k)]^T$ contains the source signals sampled at $1 \le k \le K$ and the matrix $\boldsymbol{A}$ called mixing matrix is the transfer function between sources and sensors. The source signals $s_i(k)$, $1 \le i \le M$ ($M \square N$), are assumed independent.

To obtain source signals from observations one utilizes *Blind Sources Separation* (*BSS*) algorithm entitled *Robust Second Order Blind Identification* (*RSOBI*). This one consists of an orthogonalization stage fallowed by a unitary transform.

Orthogonalization stage is performed by *Robust orthogonalization* algorithm described by A. Cichocki [6]. For preselected delays (p_1, p_2, ... , p_J) one estimates a set of symmetric delayed covariance matrices of sensor signals:

$$\tilde{\boldsymbol{R}}_x(p_j) = \frac{\boldsymbol{R}_x(p_J) + \boldsymbol{R}_x^T(p_J)}{2},\ j = 1,\ldots, J \tag{2}$$

where $\boldsymbol{R}_x(p)$ is the delayed covariance matrix of the observation vector computed as:

$$\boldsymbol{R}_x(p) = E\left[\boldsymbol{x}(k)\boldsymbol{x}^T(k-p)\right] = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{x}(k)\ \boldsymbol{x}^T(k-p) \tag{3}$$

and one constructs an NxNJ matrix:

$$\boldsymbol{R} = [\,\tilde{\boldsymbol{R}}_x(p_1)\ ,\ \ldots\ ,\ \tilde{\boldsymbol{R}}_x(p_J)\,] \tag{4}$$

Then it is performed a singular values decomposition of matrix $\boldsymbol{R}$:

$$\boldsymbol{R} = \boldsymbol{Q}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^T \tag{5}$$

where NxM matrix $\boldsymbol{Q}$ and NJxNJ matrix $\boldsymbol{W}$ are orthogonal, and $\boldsymbol{\Sigma}$ is an MxNJ matrix whose left M columns contain $diag[\sigma_1\tilde{\sigma}_{\square}\sigma_{\square}]$ (with non increasing singular values) and whose right NJ-M columns are zero. For a non-zero initial vector of parameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2 ,\ldots, \alpha_J]^T$ one computes the linear combination:

$$\overline{\boldsymbol{R}} = \sum_{j=1}^{J}\alpha_j\ \boldsymbol{Q}^T\ \tilde{\boldsymbol{R}}_x(p_j)\boldsymbol{Q} = \sum_{j=1}^{J}\alpha_j\boldsymbol{R}_j \tag{6}$$

One checks if $\overline{\boldsymbol{R}}$ is positive definite ($\overline{\boldsymbol{R}}$>0) and one performs the eigenvalues decomposition of $\overline{\boldsymbol{R}}$. If $\overline{\boldsymbol{R}}$ isn't positive definite one chooses an eigenvector $\boldsymbol{v}$ corresponding to the smallest eigenvalue of $\overline{\boldsymbol{R}}$ and one updates $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha}+\boldsymbol{\delta}$, where

$$\boldsymbol{\delta} = \frac{\left[\boldsymbol{v}^T\boldsymbol{R}_1\boldsymbol{v}\ \ldots\ \boldsymbol{v}^T\boldsymbol{R}_J\boldsymbol{v}\right]^T}{\left\|\left[\boldsymbol{v}^T\boldsymbol{R}_1\boldsymbol{v}\ \ldots\ \boldsymbol{v}^T\boldsymbol{R}_J\boldsymbol{v}\right]\right\|} \tag{7}$$

and with new vector $\boldsymbol{\alpha}$ one returns to compute the linear combination $\overline{\boldsymbol{R}}$. Otherwise, one performs the eigenvalues decomposition of symmetric positive definite matrix:

$$\overline{\boldsymbol{R}}_x(\boldsymbol{\alpha}) = \sum_{j=1}^{J}\alpha_j\ \tilde{\boldsymbol{R}}(p_j) \tag{8}$$

as follows:

$$\overline{\boldsymbol{R}}_x(\boldsymbol{\alpha}) = \boldsymbol{V}^T \boldsymbol{\Lambda}\, \boldsymbol{V} \tag{9}$$

where $\boldsymbol{\alpha}$ is the set of parameters α_i after the algorithm achieves convergence positive definiteness of the matrix $\overline{\boldsymbol{R}}$, NxM matrix $\boldsymbol{V}$ = [$\boldsymbol{v}_1$, $\boldsymbol{v}_2$, ... , $\boldsymbol{v}_M$] contains the eigenvectors corresponding to the largest M eigenvalues of $\overline{\boldsymbol{R}}$, and $\boldsymbol{\Lambda} = diag[\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M]$ contains the eigenvalues arranged in decreasing order. The *Robust orthogonalization* transformation is realized by a linear transformation with matrix $\boldsymbol{W}$:

$$\boldsymbol{y}(k) = \boldsymbol{W}\, \boldsymbol{x}(k) \tag{10}$$

where the matrix $\boldsymbol{W}$ has the form:

$$\boldsymbol{W} = \boldsymbol{\Lambda}^{-0,5}\, \boldsymbol{V}^T \tag{11}$$

The covariance matrices of the observed vector can be rewritten as:

$$\boldsymbol{R}_{\boldsymbol{x}}(p) = \boldsymbol{A}\, \boldsymbol{R}_{\boldsymbol{s}}(p)\, \boldsymbol{A}^T \tag{12}$$

Because the source signals have unit variance and are assumed to be uncorrelated, the covariance matrix of the sources vector equals the unit matrix:

$$\boldsymbol{R}_{\boldsymbol{s}}(0) = E[\boldsymbol{s}(k)\, \boldsymbol{s}^T(k)] = \boldsymbol{I} \tag{13}$$

Consequently, $\boldsymbol{R}_{\boldsymbol{s}}(p)$=$E[\boldsymbol{s}(k)\, \boldsymbol{s}^T(k - p)]$ are non-zero distinct diagonal matrices, and it follows that:

$$\boldsymbol{R}_{\boldsymbol{x}}(0) = \boldsymbol{A}\, \boldsymbol{A}^T \tag{14}$$

The components of the orthogonalized vector $\boldsymbol{y}(k)$ are mutually uncorrelated and they have unit variance. The orthogonalized covariance matrices are given by:

$$\boldsymbol{R}_y(0) = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}(k)\, \boldsymbol{y}^T(k) = \boldsymbol{W}\, \boldsymbol{R}_{\boldsymbol{x}}(0)\, \boldsymbol{W}^T = \boldsymbol{I} \tag{15}$$

$$\boldsymbol{R}_y(p) = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}(k)\, \boldsymbol{y}^T(k-p) = \boldsymbol{W}\, \boldsymbol{R}_{\boldsymbol{x}}(p)\, \boldsymbol{W}^T., p\neq 0 \tag{16}$$

From equations (15) and (16) it results:

$$\boldsymbol{R}_y(0) = \boldsymbol{W} \boldsymbol{A}\, \boldsymbol{A}^T \boldsymbol{W}^T = \boldsymbol{W} \boldsymbol{A}\, (\boldsymbol{W} \boldsymbol{A})^T = \boldsymbol{I} \tag{17}$$

Thus, it follows that $\boldsymbol{U} = \boldsymbol{W} \boldsymbol{A}$ is an N x N unitary matrix. Consequently, the determination of M x N mixing matrix $\boldsymbol{A}$ is reduced to that of a unitary N x N matrix $\boldsymbol{U}$. From equations (13) and (17) it results:

$$\boldsymbol{R}_y(p) = \boldsymbol{W} \boldsymbol{A}\, \boldsymbol{R}_s(p)\, \boldsymbol{A}^T \boldsymbol{W}^T = \boldsymbol{W} \boldsymbol{A}\, \boldsymbol{R}_s(p)\, (\boldsymbol{W} \boldsymbol{A})^T., p\neq 0 \tag{18}$$

Since $\boldsymbol{R}_s(p)$ is diagonal, any orthogonalized covariance matrix $\boldsymbol{R}_y(p)$ with $p\neq 0$ is diagonalized by the unitary transform $\boldsymbol{U}$.

The *SOBI* algorithm, developed first by A. Belouchrani [4], retrieves the unitary matrix $\boldsymbol{U}$ by jointly diagonalizing a set of delayed covariance matrices. This matrix jointly diagonalizes the set $M_{\boldsymbol{R}}$={$\boldsymbol{R}_y(p)|p$=1,..., P} when the next criterion is minimized:

$$C(M_R\,,\, \boldsymbol{U}) = \sum_{p=1}^{P} off\left(\boldsymbol{U}^T\ \boldsymbol{R}_y(p)\ \boldsymbol{U}\right) \tag{19}$$

where *off* operator is defined as:

$$off\ (M) \ = \sum_{1 \le i \ne j \le N} \left| Mij \right|^2 \qquad (20)$$

The unitary matrix $\boldsymbol{U}$ is computed as product of Givens rotations [4]. When the unitary matrix $\boldsymbol{U}$ is obtained, the mixing matrix is estimated by $\boldsymbol{A} = \boldsymbol{W}^{+} \cdot \boldsymbol{U}$ and the unmixing matrix is then given by $\boldsymbol{U}^T \boldsymbol{W}$, where + denotes the pseudo-inverse.

3. Predictive 3D Mesh Geometry Coding

The principle of the geometry encoding of single resolution 3D meshes, developed in [11] [14] [17] [20], requires the following steps: 1 the vertex coordinates are uniformly quantized and the quantization step is choosen using an iterative search algorithm in order to accomplish the bitrate control constraint; 2 a decorrelation step is performed directly by prediction on the quantized model; 3 the resulting errors are losslessly coded using successive approximations followed by arithmetic coding.

Considering a sequence of vertices $V_{i,}$ $1 \le i \le N$, each coordinate value on each direction is quantized to provide qV_i. Thanks to a linear prediction rule it is possible to obtain an estimated value $q\hat{V}_i$ of it. The difference between the current vertex V_i and its estimated $q\hat{V}_i$ is called the predictive error dV_i. Prediction technique consists of coding the predictive error only, as shown in the following figures:

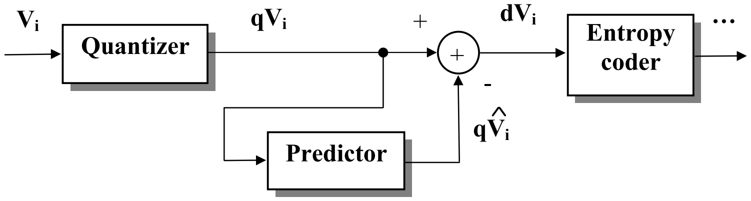


Figure 1. Encoder

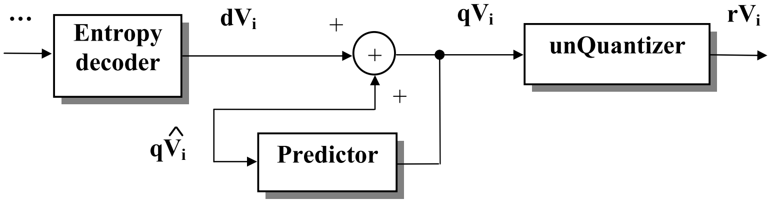


Figure 2. Decoder

4. Quantization of the Vertex Position

A cubic bounding box, defined by the minimum and maximum values of data, is first determined. The model's geometry is quantized on a uniform 3D grid, defined with regard to the bounding box and to the number of quantization levels. The quantized values are performed as follows:

$$qV \ = \ \text{int}\left(\frac{V - lC}{\max S} (2^{bpv} - 1) \right) \qquad (21)$$

where V is the floating value of the vertex on each x, y and z direction, lC is the minimum value of data on each mentioned direction, max$S$ is determined by the difference of the maximum and the minimum values for all directions and *bpv* represents the number of quantization levels. The restored floating value of the vertex, affected by the quantization error, is performed by:

$$rV = lC + \frac{\max S \cdot qV}{2^{bpv} - 1} \tag{22}$$

5. Predictive Rule

The efficiency of the compression framework is heavily based on the performances of the decorrelation step performed by the prediction rule [11].

The actual techniques of compression use the polygonal predictive rule. This one was firstly proposed by C.J. Kuo [14], and then developed as a hybrid scheme by F. Preteux and all [9]. The principle of polygonal predictive rule consists in predicting a vertex from already traversed vertices in all polygons incident to it. Consequently, let's consider a current vertex V of a regular polygon including n vertices (V_1 V_2... V_n , with n>3). In the plane, passing through three non-collinear and not necessarily consecutive vertices V_i, V_j, and V_k, the predicted value is estimated by:

$$V = \alpha(n, i, j, k, l)V_i + \beta(n, i, j, k, l)V_j + \gamma(n, i, j, k, l)V_k. \tag{23}$$

where α, β and γ are the barycentric coordinates depending only on i, j, k, l and n as follows:

$$\begin{aligned}
\alpha &= \frac{\sin\left((l-j)\frac{2\pi}{n}\right) + \sin\left((j-l)\frac{2\pi}{n}\right) + \sin\left((k-l)\frac{2\pi}{n}\right)}{\sin\left((i-j)\frac{2\pi}{n}\right) + \sin\left((j-k)\frac{2\pi}{n}\right) + \sin\left((k-i)\frac{2\pi}{n}\right)} \\
\beta &= \frac{\sin\left((i-l)\frac{2\pi}{n}\right) + \sin\left((l-k)\frac{2\pi}{n}\right) + \sin\left((k-i)\frac{2\pi}{n}\right)}{\sin\left((i-j)\frac{2\pi}{n}\right) + \sin\left((j-k)\frac{2\pi}{n}\right) + \sin\left((k-i)\frac{2\pi}{n}\right)} \\
\gamma &= \frac{\sin\left((i-j)\frac{2\pi}{n}\right) + \sin\left((j-l)\frac{2\pi}{n}\right) + \sin\left((l-i)\frac{2\pi}{n}\right)}{\sin\left((i-j)\frac{2\pi}{n}\right) + \sin\left((j-k)\frac{2\pi}{n}\right) + \sin\left((k-i)\frac{2\pi}{n}\right)}
\end{aligned} \tag{24}$$

If the current vertex belongs to a polygon including more than three predicted vertices, every possible combination of three preceding vertices yields a prediction. To make each preceding vertex contribute to the prediction of the vertex V equally, every possible combination of three preceding vertices is used to predict the vertex V. Consequently, the vertex V is predicted as the average of all these predictions. If there are several polygons incident to vertex V, which contain more than three traversed vertices, a prediction is computed for each of these polygons and the final prediction is the average of the predictions obtained from each polygon incident to V. When each polygon incident to vertex V has less than three traversed vertices, the vertex V is predicted as the mean of its traversed neighboring vertices. The first predictive error, at step i=1, is calculated as:

$$dV_1 = qV_1 \tag{25}$$

The following figure shows the prediction of vertex V, described above, where the three preceding vertices are shown in black spot:

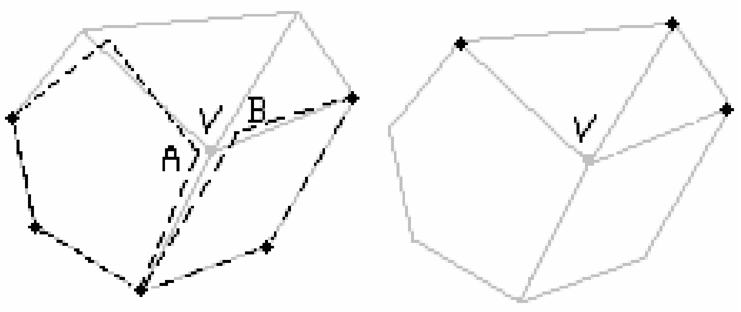


Figure 3. The prediction of vertex *V*

6. Successive Approximation and Arithmetic Coding

In order to obtain a binary representation of the input data, a successive approximation is performed [9]. This technique applies a sequence of thresholds T_i to determine if the input data are greater or not than each threshold. The thresholds are chosen so that $T_{i+1} = 0,5 \cdot T_i$, and initial threshold T_0 is set to one half of the maximum of input data. The approximations are defined by a binary string $b_0b_1b_2\ldots$ and a sign bit s.

The number of information bits for a symbol is equal to the number of bits in the optimum code for the symbol. The efficiency of transmission can be improved using shorter code words for the more probable symbols and longer code words for the less probable symbols. The principle is to associate at each symbol a number of bits that is going to depend on its probability of apparition. The binary symbols generated by the successive approximation algorithm are coded using a binary arithmetic coder with a 113 states *Markov* model for the probability estimation [15].

7. BSS Algorithms Introduction to the 3D Mesh Geometry Decorrelation

The mesh geometry is spatially correlated on each direction of the Cartesian coordinate system and its decorrelation on each axis x, y, and z allows to compress it. In order to decorrelate the geometry of 3D mesh, the linear prediction rule used by actual methods is substituted by a *Blind Sources Separation* technique. In this goal one express the vector with N components of initial geometry $\boldsymbol{g} = [V_1, V_2, \ldots, V_N]^T$ by a vector $\boldsymbol{dg} = [dV_1, dV_2, \ldots, dV_M]^T$ with M decorrelated components, associated with a mixing matrix $\boldsymbol{A}[N\text{x}M]$, where $M << N$.

Using *predictive method,* described above, the geometry $\boldsymbol{g}$ of the 3D mesh can be written as it follows:

$$\begin{cases} V_1(x,y,z) = dV_1 \\ V_2(x,y,z) = V_1 + dV_2 \\ V_3(x,y,z) = \alpha_1 V_1 + \alpha_2 V_2 + dV_3 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ V_n(x,y,z) = \alpha \cdot P_i + \beta \cdot P_j + \gamma \cdot P_k + dV_n \ , \ n \in [4, N] \end{cases} \tag{26}$$

where P_i, P_j and P_k are already crossed three vertices and α, β, γ, are the coefficients of the prediction rule mentioned by the equation (23). Replacing the P_i, P_j and P_k by their descriptions according to V_1=dV_1 and differences dV_2, dV_3, … , the geometry $\boldsymbol{g} = [V_1, V_2, \ldots, V_N]^T$ of the 3D model can be expressed thus:

$$\begin{cases} V_1(x,y,z) = dV_1 \\ V_2(x,y,z) = A_{2,1}dV_1 + dV_2 \\ V_3(x,y,z) = A_{3,1}dV_1 + A_{3,2}dV_2 + dV_3 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ V_n(x,y,z) = A_{n,1}dV_1 + A_{n,2}dV_2 + \ldots + A_{n,n-1}dV_{n-1} + dV_n \quad , \ n \in [4, N] \end{cases} \tag{27}$$

where the vector ***dg*** = $[dV_1, dV_2, \ldots, dV_N]^T$ is the decorrelated geometry of 3D mesh. One can remark that the geometry of 3D mesh is a linear combination of the decorrelated geometry's components. In this context, the geometry vector ***g*** is described by the observation vector ***x***, the three samples of its corresponding to the vertices values on the Cartesian directions:

$$\begin{pmatrix} x_1(1) & x_1(2) & x_1(3) \\ x_2(1) & x_2(2) & x_2(3) \\ \ldots & \ldots & \ldots \\ x_N(1) & x_N(2) & x_N(3) \end{pmatrix} = \begin{pmatrix} V_1(x) & V_1(y) & V_1(z) \\ V_2(x) & V_2(y) & V_2(z) \\ \ldots & \ldots & \ldots \\ V_N(x) & V_N(y) & V_N(z) \end{pmatrix} \tag{28}$$

This assumption leads us to conclude that the recovered sources vector ***s*** using *BSS* method approximately equal the decorrelated geometry vector ***dg*** that was coded in the bitstream of *Predictive compression method*:

$$\begin{pmatrix} s_1(1) & s_1(2) & s_1(3) \\ s_2(1) & s_2(2) & s_2(3) \\ \ldots & \ldots & \ldots \\ s_M(1) & s_M(2) & s_M(3) \end{pmatrix} \cong \begin{pmatrix} dV_1(x) & dV_1(y) & dV_1(z) \\ dV_2(x) & dV_2(y) & dV_2(z) \\ \ldots & \ldots & \ldots \\ dV_M(x) & dV_M(y) & dV_M(z) \end{pmatrix} \tag{29}$$

and that the mixing matrix ***A*** is represented by:

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ A_{2,1} & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ A_{M,1} & A_{M,2} & \ldots & 1 \end{pmatrix} \tag{30}$$

Since the neighbors of the current vertex, that were determined during the passage on the triangle tree in the encoding process of the connectivity [18], are the same indifferent of *x*, *y* or *z* considered coordinate, the coefficients α, β and γ are not different in the three situations (α_x=α_y=α_z, β_x=β_y=β_z, γ_x=γ_y=γ_z). This notice explains the uniqueness of the matrix ***A*** (***A****ₓ*=***A****ᵧ*=***A****_z*).

8. 3D Mesh Geometry Coding by *BSS*

The suggested compression technique based on *BSS* algorithms, decorrelates the 3D mesh geometry, preserving the information needed for the reconstruction in the reduced mixing matrix (*N*x*M* in dimension) and in the extracted sources with three samples corresponding to the *Cartesian* coordinates *x*, *y*, and *z*. The numbers of extracted sources being very small, their values are included in the compressed file's header. The mixing matrix, having uncorellated elements, makes that the corresponding binary data contain more identification bits and less refinement bits. The matrix elements are quantified, binaried 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.

It is to be mentioned that the number of data hasn’t decreased after processing. On the contrary, besides the matrix elements in the number of *N*x*M*, also came up the values of the extracted sources on the directions *x*, *y*, and *z*. The advantage obtained after the process through the *Blind Sources Separation* method must be searched for somewhere else such as: seeing these through the compression angle, after processing it goes from the spatially correlated geometry data to the mixing matrix data that don’t have the same property any longer. In contrast to the correlated geometry status, the binary data resulted 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.

At the reception it is achieved the geometry reconstruction through the mixing of decorrelated geometry (the sources extracted at emission) using the arithmetic decoded and dequantized mixing matrix.

The *M* lines of the mixing matrix have an equal dimension to the one of the original mesh geometry. With the purpose of reducing this deficiency, which aggravates 3D mesh geometry compression, we suggest the geometry division into blocks ***bg*** with *N*/*b* dimension (with the rounding towards the superior unit if the previous ratio is a rational number), where *b* is the number of blocks:

$$\boldsymbol{bg}(i) = \boldsymbol{g}\left((i-1)\frac{M}{b} + 1 \; : \; i\,\frac{M}{b} \right) \tag{31}$$

This measure was tested for *Karhunen-Loève Transfom* [8] and *Eigenvalues Decomposition* [7]. Now it is update in association with *Robust Second Order Blind Identification* algorithm.

For each block there is determined a set of delayed covariance matrices $S_i=\{\boldsymbol{R}_{x_i}(p_j), j=1,...,J\}$ (i=1,...,b). Next it is calculated a global set of matrices, in which each matrix represents the average of the corresponding matrices $\boldsymbol{R}_{x_i}(p_j)$ (i=1,...,b) from each set of matrices suitable to the blocks. This global set of matrices is used at the calculation of the symmetric positive defined matrix, whose eigenvalues decomposition determines the matrix $\boldsymbol{W_b}$ of the orthogonalization transform. The $\boldsymbol{W_b}$ matrix is applied to each block of the geometry:

$$\underline{\boldsymbol{bg}}[M\text{x}3] = \boldsymbol{W_b}[M\text{x}\frac{N}{b}] \cdot \boldsymbol{bg}[\frac{N}{b}\text{x}3] \tag{32}$$

For each block orthogonalized geometry there is determined a set of delayed covariance matrices $M_i=\{\boldsymbol{R}_i(\tau_p), p=1,...,P\}$ (i=1,...,b). Next it is calculated a global set of matrices, in which each matrix represents the average of the corresponding matrices $\boldsymbol{R}_i(\tau_p)$ (i=1,...,b) from each set of matrices suitable to the blocks. The jointly approximate diagonalisation of the global set of the matrices determines a global unitary matrix $\boldsymbol{U_b}$, used to separate the sources. By applying to each block of the whitened geometry $\underline{\boldsymbol{bg}}$ the transposed of the global unitary matrix, it is obtained the sources separation:

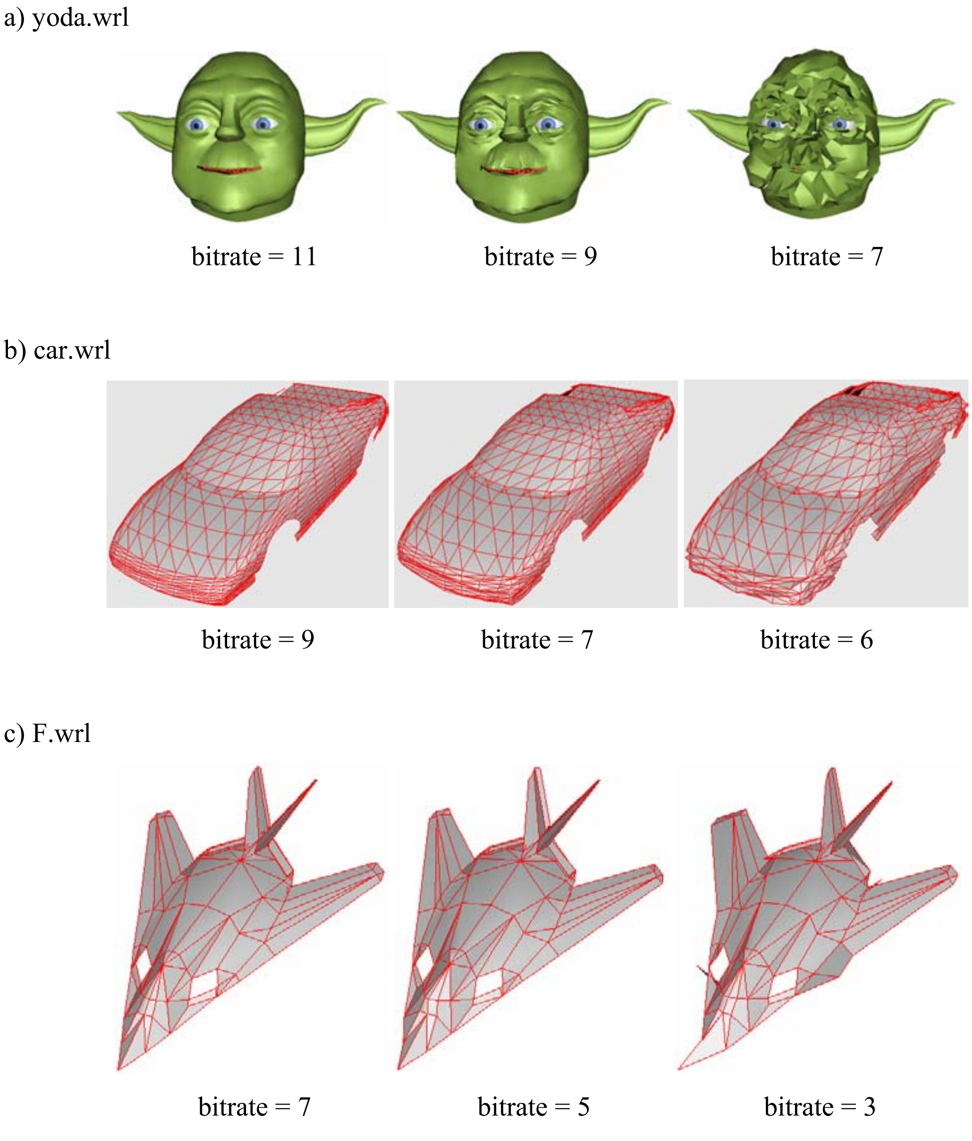
$$\boldsymbol{bdg}[M\text{x}3] = \boldsymbol{U_b^T}\,[M\text{x}M] \cdot \underline{\boldsymbol{bg}}[M\text{x}3] \tag{33}$$

Thus, the mixing matrix dimension decreases from *M*x*N* to *M*x*N/b* and the decorrelated geometry dimension increases from *M* to *b·M*. A fine choice of blocks number in which the correlated geometry vector is divided, leads to decrease the mixing matrix dimension and to increase decorrelated geometry vector a little. This result brings a good decorrelation and compression, with small errors at reconstruction. If one chooses a too large number of blocks for decreasing the global mixing matrix dimension as much as possible, one obtains an increase of the reconstruction error. The increase of number *M* of considered components from transformed geometry, for obtaining a decrease of reconstruction error, provides increase of mixing matrix dimension by its rows number and of decorrelated geometry dimension, which yields a small mesh compression.

9. Experimental Results

The compression technique based on *BSS* algorithm has been applied on several VRML models.

Results were evaluated subjectively with a VRML browser, by visual inspection of the reconstructed meshes for a quality impression and objectively by a measure of distortions named reconstruction error. In the next figure the reconstruction results for three representative VRML models are illustrated. The mesh reconstruction has been affected by both the successive approximation error and the error corresponding to neglected coefficients with small energy. The first image corresponds to a good reconstruction for which the bitrate is optimum and the reconstruction error is small. In the last image one can see the degradation by using a too small bitrate. The number of bits used at binary representation is called bitrate.


a) yoda.wrl

bitrate = 11 bitrate = 9 bitrate = 7

b) car.wrl

bitrate = 9 bitrate = 7 bitrate = 6

c) F.wrl

bitrate = 7 bitrate = 5 bitrate = 3

Figure 3. Reconstructed VRML models

The distortion used in this study is defined as the distance between the vertices from the original and reconstructed mesh. The measure of distortions is defined as the mean value of these distortions for all vertices. Consequently, for objective evaluation the reconstruction error was computed as:

$$e_r = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i\text{-}rx_i)^2 + (y_i\text{-}ry_i)^2 + (z_i\text{-}rz_i)^2} \tag{34}$$

where *N* denotes the number of vertices, x_i, y_i and z_i are the original values of geometry and rx_i, ry_i and rz_i are the values of reconstructed geometry.

The mesh geometry compression performances, appreciated by the geometry compression rate (*r*[%]) and reconstruction error corresponding to some values of bitrate, are shown in the next table. Every line of the table corresponds to the same reconstruction error approximately. Geometry compression rate is:

$$r = \frac{so - sc}{so}\cdot 100[\%] \tag{35}$$

where *so* and *sc* represent the size of the original respectively compressed file.

Table 1. b/v = bitrate [bits/vertex]; *r* = geometry compression rate

yoda.wrl		car.wrl		F.wrl	
b / v	*r*[%]	b / v	*r*[%]	b / v	*r*[%]
11	86	9	86	7	85
10	89	8	89	6	88
9	91	7	91	5	90
7	93	6	92	3	92

6. Conclusion

Starting from the vertices coordinates decorrelation obtained after using a linear prediction law we have shown that the 3D mesh geometry is a linear combination by the decorrelated geometry vector's components. This remark allows the applying of the *Blind Sources Separation* algorithms to the compression of mesh geometry of virtual reality.

Based on the experimental results we conclude that the best results for compression and reconstruction of 3D meshes are obtained when the correlated geometry is divided in two blocks (*b*=2), one separated three sources (*M*=3) from each block of correlated geometry, and one used a global reconstruction matrix with dimension equals *N/*2 x 3. The resulted error in mesh reconstruction is small. It is aproximativelly equal to the obtained error when the correlated geometry vector is not divided and one separated three sources. But the new dimension of reconstruction matrix provides good geometry compression.

The obtained results are comparable with those of actual compression methods of 3D meshes, *BSS* algorithms offering an alternative way.

REFERENCES

1. AMARI, S., J. F. CARDOSO, **Blind Source separation – semiparametric statistical approach**, IEEE Trans. Signal Processing, 45(11): 2692-2700.
2. BAJAJ, C.L., V. PASCUCCI, G. ZHUANG, **Single Resolution Compression of Arbitrary Triangular Meshes with Properties**, Technical report, TICAM, The University of Texas at Austin, 1998.
3. BELL, T., T. SEJNOWSKY, **An information maximization approach to blind separation and blind deconvolution**, Neural Computation, vol. 7, 1995, pp. 1129-1159.
4. BELOUCHRANI, A., K. ABDEL-MERAIM, J.F. CARDOSO, E. MOULINES, **A blind source separation technique using second-order statistics**, IEEE Trans. Signal Processing, vol. 45, 1997, pp. 434-444.
5. CHOW, M., **Optimizel Geometry Compression for Real-time Rendering**, In Proceedinngs of IEEE Visualization '97, Phowenix AZ, 1997, pp. 347-354.
6. CICHOCKI, A., 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. CURILĂ, M., D. NUZILLARD, S. CURILĂ, **Séparation Aveugle de Sources appliquée à la compression du maillage 3D**, 19st GRETSI Symposium on Signal and Image Processing, Paris, France, 8-11 september, 2003.
8. CURILĂ, M., S. CURILĂ, D. NUZILLARD, **Decorrelation techniques for geometry coding of 3D mesh**, Fourth International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, Avril 1-4, 2003, pp. 885-890.
9. CURILĂ, S., M. CURILĂ, T. ZAHARIA, G. MOZELLE, F. PRETEUX, A **new prediction scheme for geometry coding of 3D meshes within the MPEG-4 framerwork**, Nonliniear Image Processing X, Proceedings of SPIE, '99, San Jose, California, January, pp. 240-250, 1999.
10. DEERING, M., **Geometric Compression**, Computer Graphics (SIGGRAPH '95 Proceedings), 1995, pp. 13-20.
11. PRETEUX, F., T. ZAHARIA, M. CURILĂ, S. CURILĂ, G. MOZELLE, **Geometry compression of 3D meshes: Results on Core Experiment M2**, Report ISO/IEC JTC1/SC29/WG11, MPEG97/M4058, Atlantic City, October 1998.
12. GUÉZIEC, A., G. TAUBIN, F. LAZARUS W. HORN, **A Framework for Streaming Geometry in VRML**, IEEE Computer Graphics and Applications. 19(2), march-April 1999, pp. 68-78.
13. GUEZIEC, A., F. BOSSEN, G. TAUBIN, C. SILVA, **Efficient Compression of Non-Manifold Polygonal Meshes**, IEEE Visualization '99, 1999.
14. LI, J., C. J. KUO, **Embedded Coding of Mesh Geometry**, Research Report ISO/IEC JTC1/SC29/WG11, MPEG98 / M3325, Tokyo, Japan, March, 1998.
15. MOFFAT, A., R. NEAL, I. WITTEN, **Arithmetic Coding Revisited**, In IEEE Data Compression Conference, Snowbirb, 16(3), Utah, 1995, pp. 256-294.
16. NUZILLARD, D., J.-M. NUZILLARD, **Second order blind source separation on the fourier space of data**, To be published in: Signal Processing 2003.
17. TAUBIN, G., J. ROSSIGNAC, **Course on 3D Geometry Compression**, Siggraph '2000, New Orleans, Louisiana, July 2000.

18. TAUBIN, G., W.P. HORN, F. LAZARUS, and J. ROSSIGNAC, **Geometric Coding and VRML**, Proceedings of the IEEE. 86(6), June 1998, pp 1228-1243.

19. TAUBIN, G., J. ROSSIGNAC, **Geometric Compression through Topological Surgery**, ACM Transaction on Graphics, April 1998, pp. 84-115.

20. TAUBIN, G., **3D Geometry Compression and Progressive Transmission**, Eurographics State of the Art Report, September 1999.

21. TAUBIN G., and J. ROSSIGNAC, **Course on 3D Geometry Compression**, Siggraph'99, Los Angeles, CA, August 1999.

22. TOUMA, C., C. GOTSMAN, **Triangle mesh compression**, In Graphics Interface Conference Proceegings, Vancouver, June, 1998, pp. 26-34.

23. The Virtual Reality Modeling Language, ISO / IEC 14772 - 1, 1997, http://www.vrml.org/Specifications//VRML97.