DECORRELATION TECHNIQUES FOR GEOMETRY CODING OF 3D MESH

Mircea Curila **, Sorin Curila* **, Danielle Nuzillard**
Laboratoire d'Automatique et de Microélectronique*, Université de Reims Champagne-Ardenne, Campus du Moulin de la Housse
B.P. 1039, 51687 REIMS Cedex 2, FRANCE, Tel: +33.(0)3.26.91.82.16, Fax : +33.(0)3.26.91.31.06
University of Oradea**, 3700 Oradea, ROMANIA, str. Armatei Romane 5
e-mail: mircea.curila@univ-reims.fr, sorin.curila@univ-reims.fr, danielle.nuzillard@univ-reims.fr

ABSTRACT

There exists a lot of redundancy in 3D meshes that can be exploited by Blind Source Separation (BSS) and Independent Component Analysis (ICA) for the goal of mesh compression. The 3D meshes geometry is spatially correlated on each direction in the Cartesian coordinate system. In the context of mesh compression with BSS algorithm, it is proposed to take the correlated geometry of 3D mesh as observations and decorrelated geometry as sources corresponding at largest energy. The geometry decorrelation is obtained by using *EVD*, *SOBI*, and *Karhunen-Loève* transform. In *Karhunen-Loève* transform case, to reduce the transform matrix dimension, the correlated geometry is divided into blocks and then by averaging the covariance matrices of each block, a global covariance matrix is performed. The transform matrix represents the bitstream of compressed file. Its elements are quantized and encoded using arithmetic coding, providing effective compression.
Keywords - Blind Source Separation, Independent Component Analysis, Second Order Blind Identification, Eigenvalues Decomposition, Karhunen-Loève transform, Prediction, Quantization, Arithmetic coding, VRML, 3D mesh, compression.

1. INTRODUCTION

The abundance of 3D data that results from the development of multimedia systems and the exploitation of the Internet 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 [22], G. Taubin [20] [21], M. Deering [9], F. Bossen[11], H.Hoppe[12], F.Lazarus [21], M. Chow[5], C. Gostman [3], F. Preteux [18], C. Kuo [14], A.Gueziec [11], etc. In the case of 3D data represented by 3D meshes, the coding procedure involves three different coding steps for the topology, the geometry and the attributes (such as texture, color and curvature) of the mesh. Our contribution is a method concerning the geometry coding.

The Virtual Reality Modeling Language (VRML) has become the most commonly used standard for representing such 3D meshes. A VRML file contains complex information in text format related either to the topology and the geometry of the model, or to the model properties. In the VRML format, the vertex coordinates are listed first, followed by the indexed face set.

Basically, a 3D mesh is defined by a set of vertices and a set of faces. The vertex is located by its coordinates in the 3D space, usually represented in a Cartesian system. A face is defined as an ordered sequence of vertex indices. The model geometry refers to the position of the vertices, while the model connectivity information represents the relationships between vertices, edges and faces of the mesh. The model may also contain photometric information, including color, texture and normals.

2. BASIC CONCEPTS

2.1. Data model

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

$$X(k) = A\,S(k) \tag{1}$$

where the N-dimensional vector $X(k) = [X_1(k), X_2(k), \dots, X_N(k)]^T$ is an instantaneous linear mixture of source signals, the M-dimensional vector $S(k) = [S_1(k), S_2(k), \dots, S_M(k)]^T$ contains the source signals sampled at $1 \le k \le K$, the matrix A called mixing matrix is the transfer function between sources and sensors. The source signals $S_i(k)$, $1 \le i \le M$ $(M < N)$, are assumed independent.

The sources vector $S(k)$ can be considered as a stationary multivariate process:

$$E[S(k)\,S^T(k-p)] = diag[\rho_1(p), \rho_2(p), \dots, \rho_M(p)] \tag{2}$$

where *diag*[.] is the diagonal matrix formed with its arguments. Thus, the sources $S_i(k)$, $1 \le i \le M$ are mutually uncorrelated, and $\rho_i(p) = E[S_i(k)\,S_i^T(k-p)]$ represents the autocovariance of $S_i(k)$.

2.2. BSS algorithms

The goal of Blind Source Separation (BSS) is to retrieve the sources $S(k)$ from the observed data $X(k)$. Some methods in Blind Source Separation and Independent Components Analysis (ICA) use higher order statistics to obtain statistically independent sources. Others approaches such as *Karhunen-Loève* transform [13], *Eigenvalues Decomposition* [6] and *Second Order Blind Identification* [4] exploit the temporal, spatial or spectral [17] diversities of the sources. This is achieved by using delayed covariance matrices at different delays to impose a decorrelated structure on the solution.

The following subsections present the basic concepts of the tested algorithms in the experimental part. These ones consist of an orthogonalization stage followed by a normalization and a unitary transform stage.

2.2.1. Orthogonalization

Orthogonalization stage is performed by the optimum *Karhunen-Loève* transform (*KLT*) that concentrates the maximum average energy in a given number of transform coefficients while completely decorrelating them.

For observation vector, the basis vectors of the *Karhunen-Loève* transform are given by the orthogonalized eigenvectors of its covariance matrix R_X. The covariance matrix of the observation vector has the following form:

$$R_X(0) = E\left[X(k)X^T(k)\right] = \frac{1}{K}\sum_{k=1}^{K} X(k)\ X^T(k) \qquad (3)$$

Let V^T be an N x N matrix, which reduces the covariance matrix R_X to its diagonal form, that is:

$$V^T R_X\ V = \Lambda_X = diag\{\lambda_1, \lambda_{2,\dots,} \lambda_N\} \qquad (4)$$

where λ_i, $1 \le i \le N$ are eigenvalues of matrix R_X. The rows of orthogonal matrix V^T are the transposed eigenvectors of the covariance matrix R_X.

The transformed vector:

$$Y = V^T X \qquad (5)$$

has orthogonal and decorrelated elements Y_i, $1 \le i \le N$. The inverse transform, that recovers vector X from the transformed vector Y is:

$$X = V\,Y = \sum_{i=1}^{N} Y_i\ V_i \qquad (6)$$

where V_i is the *i*-th column of matrix V, and V is the inverse of the matrix V^T, because V is orthogonal matrix.

The dimension of the vector Y can be reduced by neglecting eigenvectors that correspond to small eigenvalues. If one considers the matrix obtained by neglecting last N - M rows of matrix V^T, where the eigenvectors were arranged corresponding to a decreasing order of eigenvalues ($\lambda_1 \ge \lambda_2 \ge \dots \ge \lambda_N$), the dimension of transformed vector Y becomes M x 1, where $M << N$. Thus, only the coefficients corresponding to the highest energy are retained. The inverse transform gives a reconstruction of the original vector with loss.

The reconstruction mean square error:

$$\sigma_e^2 = \sum_{k=M+1}^{N} \lambda_k \qquad (7)$$

is equal to eigenvalues sum corresponding to the neglected eigenvectors. Generally, the eigenvalues have significant dimension, therefore neglecting the small eigenvalues introduces insignificant errors. As it will be seen with a VRML browser, retaining all the coefficients is not necessary for a good quality of reconstructed 3D mesh.

When the input vector elements are highly correlated, like the geometry of 3D mesh on each coordinate axe, the transformed vector elements tend to be uncorrelated.

2.2.2. Normalization

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

$$R_X(0) = A\ R_S(0)\ A^T \qquad (8)$$

$$R_X(p) = A\ R_S(p)\ A^T,\ p \ne 0 \qquad (9)$$

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

$$R_S(0) = E[S(k)\ S^T(k)] = I \qquad (10)$$

Consequently, both $R_S(0) = E[S(k)\ S^T(k)]$ and $R_S(p) = E[S(k)\ S^T(k - p)]$ are non-zero distinct diagonal matrices, and it follows that:

$$R_X(0) = A\ A^T \qquad (11)$$

Next, the whitening is realized by a linear transformation with matrix W computed so that the whitened covariance matrix $R_Y(0)$ becomes the unit matrix I:

$$W = \Lambda_S^{-0,5}\ V_S^T \qquad (12)$$

where N x M matrix $V_S = [V_1, V_2, \dots, V_M]$ contains the eigenvectors corresponding to the largest M eigenvalues of $R_X(0)$, and $\Lambda_S = diag[\lambda_1 \ge \lambda_2 \ge \dots \ge \lambda_M]$ is arranged in decreasing order. Hence, the whitened observations are given by:

$$Y(k) = W\,X(k) = \Lambda_S^{-0,5}\ V_S^T\ X(k) \qquad (13)$$

Thus, the components of the whitened vectors $Y(k)$ are mutually uncorrelated and they have unit variance. The spatially whitened covariance matrix $R_Y(0)$ is given by:

$$R_Y(0) = \frac{1}{K}\sum_{k=1}^{K} Y(k)\,Y^T(k) = W\ R_X(0)\ W^T = I \qquad (14)$$

Two alternative methods that perform the unitary transform are available: *Eigenvalues Decomposition (EVD)* [7] and *Second Order Blind Identification (SOBI)* [5].

2.2.3. Eigenvalues Decomposition (EVD)

In order to estimate the mixing matrix A, Cichocki [6] performs simultaneous diagonalization of two covariance matrices $R_X(0)$ and $R_Y(p)$, $p \ne 0$, which consists to diagonalize simultaneously two matrices only. It is performed by an orthogonalization step followed by a unitary transformation.

The delayed covariance matrix resulted by the normalization stage described above is given by:

$$R_Y(p) = \frac{1}{K}\sum_{k=1}^{K} Y(k)Y^T(k-p) = W\, R_X(p)\, W^T, \; p \neq 0 \quad (15)$$

From equations (11) and (14), respectively (9) and (15) it follows:

$$R_Y(0) = W\,A\,A^T\,W^T = W\,A\,(W\,A)^T = I \quad (16)$$

$$R_Y(p) = W\,A\,R_S(p)\,A^T\,W^T = W\,A\,R_S(p)\,(W\,A)^T, p \neq 0 \quad (17)$$

From equation (16) it follows that $U = W\,A$ is a N x N unitary matrix. Consequently, the determination of M x N mixture matrix A is reduced to that of a unitary N x N matrix U.

Next, an orthogonal transformation is applied to diagonalize the matrix $R_Y(p)$, $p \neq 0$. An eigenvalues decomposition of whitened covariance matrix $R_Y(p)$ is performed such as:

$$R_Y(p) = V_Y\,\Lambda_Y\,V_Y^T \quad (18)$$

If the diagonal matrix Λ_Y has both distinct and sufficiently far away eigenvalues, the mixing matrix A can be estimated uniquely:

$$A = W^{1}\,V_Y = V_S\,\Lambda_S^{0,5}\,V_Y \quad (19)$$

The source signals are obtained by:

$$S(k) = V_Y^T\,Y(k) = V_Y^T\,\Lambda_S^{-0,5}\,V_S^T\,X(k) \quad (20)$$

2.2.4. SOBI algorithm

SOBI algorithm was developed by Belouchrani [4]. The first step consists of whitening the signal $X(k)$ of the observations described above. The second step retrieves the unitary matrix U by jointly diagonalizing a set of delayed covariance matrices.

Since $R_S(p)$ is diagonal, any whitened covariance matrix $R_Y(p)$ is diagonalized by the unitary transform U. This matrix jointly diagonalizes the set $M_R=\{R_Y(p)|p=1,\ldots, P\}$ when the next criterion is minimized:

$$C(M_R,V) = \sum_{p=1}^{P} off\,(V^T\,R_Y(p)\,V) \quad (21)$$

where *off* operator is defined as:

$$off\,(M) = \sum_{1\le i \neq j \le N} |Mij|^2 \quad (22)$$

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

2.3. Predictive geometry coding of 3D mesh

The principle of the geometry encoding of single resolution 3D meshes, was developed in [8] [14] [18] [20], it 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 residues are losslessly coded using successive approximations followed by arithmetic coding.

The efficiency of the compression framework is heavily based on the performances of the decorrelation step performed by the prediction technique [8]. Considering a sequence of vertices $Vertex_i$, $1 \le i \le N$, each coordinate value on each direction is quantized to provide $qVertex_i$. The difference between the current vertex $Vertex_i$ and its estimated $qVer\hat{t}ex_i$ is called the predictive error $dVertex_i$. Prediction technique consists of coding the prediction error only.

When $qVertex_i$ becomes the current vertex, at encoding step i, let $qVertex_{i-1}$, $qVertex_{i-2}$, ... $qVertex_s$ be the last s quantized values. The estimated value $qVer\hat{t}ex_i$ of $qVertex_i$ is determined using a linear predictive rule P [20]:

$$qVer\hat{t}ex_i = P(qVertex_{i-1}, qVertex_{i-2}, \ldots, qVertex_s) \quad (23)$$

The differences that will be coded in the bitstream are expressed by:

$$dVertex_n = qVertex_n - qVer\hat{t}ex_n \quad (24)$$

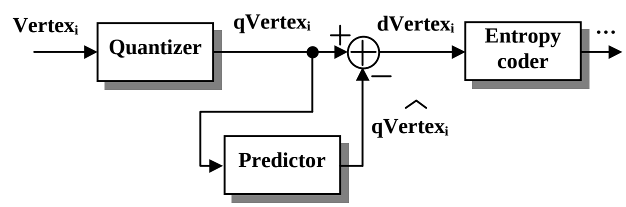

Fig. 1. Encoder

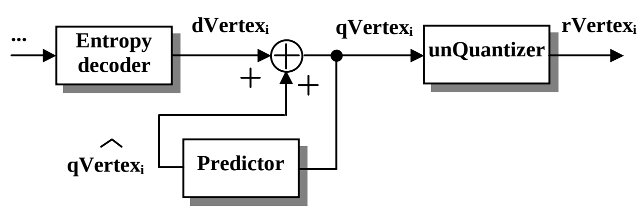

Fig. 2. Decoder

The first prediction error, at step $i = 1$, is calculated as:

$$dVertex_1 = qVertex_1 \quad (25)$$

2.4. Quantization of floating point data

A cubic bounding box, defined by the minimum and maximum values of data, is first determined. The floating point data of the model is uniformly quantized on a 3D grid, defined with regard to the bounding box and to the number of quantization levels. Using more sophisticated quantization techniques, like adaptive quantizations, generally leads to a high computational complexity, which is not justified by the gain in compression efficiency in comparison with the use of a uniform scalar quantization. The quantized values are performed as follows:

$$qVertex = \text{int}\left(\frac{Vertex - lC}{\max S}(2^{bpv} - 1)\right) \qquad (26)$$

where *Vertex* is the floating point data 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 point values, affected by the quantization error, are performed by:

$$rVertex = lC + \frac{\max S \cdot qVertex}{2^{bpv} - 1} \qquad (27)$$

2.5. Arithmetic coding

Information and entropy concepts are very important in data compression. 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. Entropic coders use a "statistical model" to convert descriptors into symbols. The descriptors used by the arithmetic coder, that is a very efficient entropic coder, are the probabilities of the symbols. If for a given set of symbols, probabilities do not change (i.e. stationary symbols) the entropic coder provides a lower bond of the code. The un-stationary symbols, like as refinement and sign bits, are coded using a fixed probability model with equal probabilities. The binary symbols generated by the quantization algorithm are coded using a binary arithmetic coder with 113 states *Markov* model for the probability estimation. The vertices are scanned one by one and, for a given vertex, its entire binary representation is coded. The next vertex is processed in the same way.

3. PROBLEM FORMULATION

As can be deduced from the reconstructed 3D mesh, there exists a lot of redundancy that can be exploited for the purpose of mesh compression by methods such as Blind Source Separation (BSS). Indeed the mesh geometry is spatially correlated on each direction of the Cartesian coordinate system and its decorelation on each axis x, y, and z allows to compress it.

The goal is to express an initial set of data $geometry = [Vertex_1, Vertex_2, \dots, Vertex_N]^T$ by a smaller set of data $dgeometry = [dVertex_1, dVertex_2, \dots, dVertex_M]^T$ and a mixing matrix $A[N\text{x}M]$, where $M << N$.

It is easy to show that the components of the geometry vector: $geometry = [Vertex_1, Vertex_2, \dots, Vertex_N]^T$ of 3D mesh, using the *Predictive geometry coding of 3D mesh*, can be written as follows:

$$\begin{cases} Vertex_1 = Vertex_1 \\ Vertex_2 = A_{2,1}\, Vertex_1 + dVertex_2 \\ \dots\dots \\ Vertex_N = A_{N,1}\, Vertex_1 + A_{N,2}\, dVertex_2 + \dots + dVertex_N \end{cases} \qquad (28)$$

where the vector $dgeometry = [qVertex_1, dVertex_2, \dots, dVertex_N]^T$ is the decorrelated geometry of 3D mesh. One can remark that the geometry of 3D mesh is a linear combination of the decorrelated geometry. In this context, the geometry vector is the observation vector:

$$[X_1(k), X_2(k), \dots, X_N(k)]^T = [Vertex_1(x, y, z), Vertex_2(x, y, z), \dots, Vertex_N(x, y, z)]^T \qquad (29)$$

where k = 1, 2, 3 for respectively x, y, z directions of the Cartesian coordinate system. This assumption leads us to conclude that the recovered sources vector using BSS method approximately equal the decorrelated geometry vector that was coded in the bitstream of *Predictive compression method*:

$$[S_1(k), S_2(k), \dots]^T = [dVertex_1(x, y, z), dVertex_2(x, y, z), \dots]^T \qquad (30)$$

and that the mixing matrix A is represented by:

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 \\ A_{2,1} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_{N,1} & A_{N,2} & \dots & 1 \end{pmatrix} \qquad (31)$$

To decorrelate the input data into maximum possible extend three algorithms were tested: *Eigenvalues Decomposition (EVD)* [6], *Second Order Blind Identification (SOBI)* [4], and *Karhunen-Loève Transform* [13]. Cartesian coordinates of the 3D mesh vertices provide three N-dimensional vectors *X*, *Y*, and *Z,* which are simultaneously processed by *EVD* and *SOBI* methods, and by *Karhunen-Loève* transform successively.

In the *Karhunen-Loève* transform case the original matrix transform has on its rows the transposed eigenvectors, which have the same dimension as the original mesh geometry. To decrease this disadvantage, that makes difficult the mesh geometry compression, the geometry vector is divided into blocks with N / b dimension, or approximate it with superior integer if previous ratio is not an integer, where b represent the number of the blocks:

$$block\_geometry(i) = geometry((i-1)\frac{N}{b} + 1 : i\frac{N}{b}) \qquad (32)$$

The covariance matrix is determined for each block, and from all of these matrices a global covariance matrix is calculated as their average. A global *Karhunen-Loève* matrix is obtained from the eigenvectors of global covariance matrix, which transforms each block:

$$block\_dgeometry[M\text{x}1] = KL\_matrix[M\text{x}\frac{N}{b}] \cdot block\_geometry[\frac{N}{b}\text{x}1] \quad (33)$$

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

4. EXPERIMENTAL RESULTS

The decorrelation techniques have been applied on several VRML models. In Fig. 3., the reconstruction results for three representative VRML models are illustrated. The required number of bits for encoding each vertex of the mesh is called bitrate. The first image corresponds to a good reconstruction for which the bitrate is optimum and the reconstruction error is small. In the second image one can see the degradation by using a too small bitrate. The mesh reconstruction has been affected by both the quantization error and the error corresponding to neglected coefficients with small energy.

Results were evaluated subjectively with a VRML browser, by visual inspection of the reconstructed meshes for a quality impression and objectively by means of a distortion measure. The measure used in this study is defined as the distance between the accordingly vertices from the original and reconstructed mesh. The distortion measure is defined as the mean value of these distortions for all vertices. Consequently, for objective evaluation the reconstruction error was computed as:

$$e_r = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(X_i\text{-}Xr_i)^2 + (Y_i\text{-}Yr_i)^2 + (Z_i\text{-}Zr_i)^2} \quad (34)$$

where N denotes the number of vertices, X_i, Y_i, and Z_i are the original values of geometry and Xr_i, Yr_i, and Zr_i are the values of reconstructed geometry.

The mesh geometry compression performances for each of the *SOBI*, *EVD* and *KLT* algorithms, appreciated on both the geometry compression rate (r[%]) and reconstruction error corresponding to some values of bitrate (e_r[%]), are shown in the following tables.

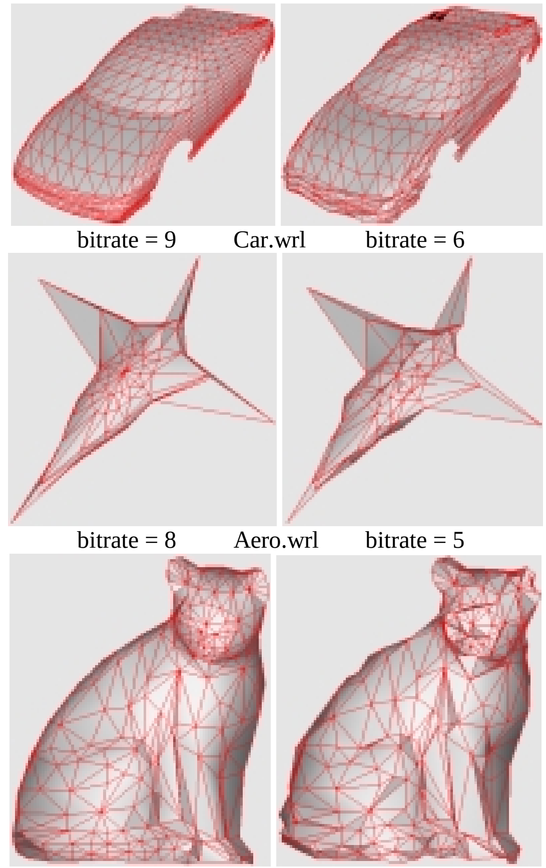

Fig. 3. Reconstructed VRML models

Geometry compression rate is the ratio between compressed geometry and original geometry of 3D mesh, which is expressed in percentages. For a correct coded model, "r" is about 80% which is valuable in data compression.

Car.wrl, number of vertices = 993

Algo.	b / v	6	7	8	9
SOBI	r[%]	84	81	78	74
	e_r[%]	3.855	1.872	0.944	0.461
EVD	r[%]	82	79	76	73
	e_r[%]	3.516	1.725	0.877	0.433
KLT	r[%]	81	78	75	72
	e_r[%]	3.041	1.526	0.749	0.372
KLTd	r[%]	91	89	88	86
	e_r[%]	2.782	1.405	0.701	0.348

Aero.wrl, number of vertices = 114

Algo.	b / v	5	6	7	8
SOBI	r[%]	83	79	76	74
	e_r[‰]	3.087	1.405	0.787	0.370
EVD	r[%]	81	78	75	71
	e_r[‰]	2.643	1.297	0.681	0.259
KLT	r[%]	83	80	77	74
	e_r[‰]	3.551	1.535	0.902	0.497
KLTd	r[%]	88	86	84	83
	e_r[‰]	1.676	0.899	0.381	0.210

Cat.wrl, number of vertices = 352

Algo.	b / v	5	6	7	8
SOBI	r[%]	83	80	77	74
	e_r[‰]	3.035	1.517	0.727	0.380
EVD	r[%]	82	79	76	72
	e_r[‰]	3.075	1.500	0.740	0.363
KLT	r[%]	83	80	77	74
	e_r[‰]	2.616	1.290	0.620	0.317
KLTd	r[%]	90	89	87	85
	e_r[‰]	2.477	1.165	0.599	0.298

The explanation of symbols used in tables is given below:
b/v = bitrate, in [bits/vertex]
r = geometry compression rate
e_r = reconstruction error
KLTd = *Karhunen-Loève* transform with geometry divided in two blocks.

5. DISCUSSION

With *EVD* and *SOBI* methods, the best results for compression and reconstruction of 3D meshes are obtained when one separated three sources from correlated geometry, and one used a reconstruction matrix with dimension equals Nx3. For *Karhunen-Loève* transform on directions x, y and z it is retained only first row of each transform matrix, which corresponds to the largest eigenvalues. In this case the best results are obtained when the correlated geometry is divided in two blocks and when a single component is considered from each block of the decorrelated geometry. 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 a single component is considered from transformed vector. The new dimension of transform matrix provides good geometry compression.

Because the components number of decorrelated geometry vector is small in proposed method, these are included into compressed file header. The transform matrix having substantial dimension, represents the bitstream of compressed file. Its elements are quantized and encoded using arithmetic coding, providing effective compression.

The obtained results are comparable with those of actual compression methods of 3D meshes. One can remark small differences between compression rates of *SOBI*, *EVD* and *KLT* algorithms, and an increase of the performance in the case of geometry divided in two blocks. All these techniques give an effective gain comparatively with original geometry described in VRML file. Our future work will be to impove the performance of these techniques.

6. REFERENCES

[1] S. Amari, J. F. Cardoso, Blind Source separation – semiparametric statistical approach, IEEE Trans. Signal Processing, 45(11): 2692-2700.

[2] C.L.Bajaj, V.Pascucci, G.Zhuang, Single Resolution Compression of Arbitrary Triangular Meshes with Properties, Technical report, TICAM, The University of Texas at Austin, 1998.

[3] R.Bar-Yehuda, C.Gostman, Time/space tradeoffs for polygon mesh rendering, ACM Transactions on Graphics, 15(2:141-152, April 1996.

[4] A.Belouchrani, K.Abdel-Meraim, J.F.Cardoso, E.Moulines, A blind source separation technique using second-order statistics, IEEE Trans. Signal Processing, vol. 45, pp. 434-444, 1997.

[5] M.Chow, Optimizel Geometry Compression for Real-time Rendering, In Proceedinngs of IEEE Visualization '97, Phowenix AZ, pages 347-354, 1997.

[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] S. Costa and S. Fiori, "Image Compression Using Principal Component Neural Networks", Image and Vision Computing Journal, Vol. 19, No. 9-10, pp. 649 - 668, Aug. 2001

[8] S.Curila, M.Curila, 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, pg. 240-250, 1999.

[9] M.Deering, Geometric Compression, Computer Graphics (SIGGRAPH '95 Proceedings), pg. 13-20, 1995.

[10] A.Guéziec, G.Taubin, F.Lazarus W. Horn., A Framework for Streaming Geometry in VRML, IEEE Computer Graphics and Applications. 19(2), pp 68-78, march-april 1999.

[11] A.Gueziec, F.Bossen, G.Taubin, C.Silva, Efficient Compression of Non-Manifold Polygonal Meshes, IEEE Visualization'99, 1999.

[12] H.Hoppe, Progressive Meshes, Computer Graphics (Proc. SIGGRAPH), pp. 99-108, August 1996.

[13] A.K.Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1988.

[14] J.Li, C.J.Kuo "Embedded Coding of Mesh Geometry", Research Report ISO/IEC JTC1/SC29/WG11, MPEG98 / M3325, Tokyo, Japan, March, 1998.

[15] A.Moffat, R.Neal, I.Witten, Arithmetic Coding Revisited, In IEEE Data Compression Conference, Snowbirb, Utah, 1995.

[16] D. Nuzillard, A. Bijaoui , Blind Source separation and Analysis of multispectral Astronomical Images, Astronomy & Astrophysics, Sup. Series, vol. 147, pp. 129-138, 2000.

[17] D. Nuzillard, J.-M. Nuzillard, Second order blind source separation on the fourier space of data, To be published in: Signal Processing 2003.

[18] F. Preteux, T. Zaharia, S. Curila, M. Curila, Geometry coding of 3D meshes: Results of Core Experiment M2, Research Report ISO/IEC JTC1/SC29/WG11, MPEG97/M4277, Roma, Italy, December 1998.

[19] A.T. Puga, A.P. Alves, An Experiment on Comparing PCA and ICA in Classical Transform Image Coding, First International Workshop on Independent Component Analysis and Signal Separation (ICA'99), Aussois, pp. 105-108, January, 1999.

[20] G.Taubin, J.Rossignac, Course on 3D Geometry Compression, Siggraph'2000, New Orleans, Louisiana, July 2000.

[21] G.Taubin, W.P.Horn, F.Lazarus, and J.Rossignac, Geometric Coding and VRML, Proceedings of the IEEE. 86(6), pp 1228-1243, June 1998.

[22] G.Taubin, J.Rossignac, Geometric Compression through Topological Surgery, ACM Transaction on Graphics, pp. 84-115, April 1998.