New spectral decomposition for 3D polygonal meshes and its application to watermarking

Kohei MUROTANI and Kokichi SUGIHARA
Department of Mathematical Informatics Graduate School of Information Science and Technology University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
muro@simplex.t.u-tokyo.ac.jp and sugihara@mist.i.u-tokyo.ac.jp

Abstract

This paper present a generalization of a data analysis technique called a singular spectrum analysis (SSA). The original SSA is a tool for analyzing one-dimensional data such as time series, whereas our generalization is suitable for multi-dimensional data such as 3D polygonal meshes. One of applications of the proposed generalization are also shown. The application of the generalized SSA is a new robust watermarking method that adds a watermark to a 3D polygonal mesh. Watermarks embedded by our method are resistant to similarity transformations and random noises. Our method has the advantage in that it requires smaller calculation cost than other methods with nearly equal performance.

Keywords: singular spectrum analysis (SSA), watermarking, 3D polygonal meshe, spectral decomposition

1 Introduction

Techniques of the spectrum decomposition have been developed in various fields, such as signal processing and financial data analysis. The Fourier analysis and the wavelet analysis are the most frequently used techniques of the spectrum decomposition. However, since they are the decompositions by a certain basis functions, the data should be represented in a parameterized form. They are applied to one-dimensional series or the tensor product of one-dimensional series naturally, while they cannot be applied to non-parameterized data. For example, we cannot represent an unbounded two-manifold having the same topology as a sphere by two parameters. In this case, for example, we can perform the spectrum decomposition using the spherical harmonics. But, if this spectrum decomposition is performed for a piecewise linear function, such as a 3D polygonal mesh, the numerous terms are required. Since spectrum decompositions by hitherto known functions have limitations like this, there is a real need for new spectrum decomposition methods.

Conference proceedings ISBN 80-903100-7-9
WSCG '2005, January 31-February 4, 2005
Plzen, Czech Republic.

We generalize the singular spectrum analysis (SSA) in such a way that it is applicable to the 3D polygonal mesh, and apply it to engineering problems [6, 8, 9, 10]. In this paper, the generalized SSA is applied to a robust watermarking method that adds a watermark to a mesh.

2 Generalization of singular spectrum analysis

2.1 From basic SSA to generalized SSA

Since the basic singular spectrum analysis (SSA) [2, 3, 17] is designed for the analysis one dimensional sequences such as time series, it is not appropriate for the 3D polygonal mesh. In this section, we generalize the basic SSA in such a way that it can be applied to the analysis of multi-dimensional data such as polygonal meshes [9].

2.2 Generalized SSA

Let the elements of the series F be the values given to the vertices of the mesh. In the case of the 3D polygonal mesh, each element of the series F consists of the coordinates of a vertex in the mesh, i.e. the position vector. For simplicity, we consider the mesh specified by the heights of vertices of a graph on the plane as shown in Figure 1. Let the elements of the series F be the heights given to the vertices of the graph.

Let N be a positive (usually large) integer. Consider a real-value series $F = (f_0, f_1, \ldots, f_{N-1})$ of length N. Assume that F is a nonzero series, that is, there exist at least one i such that $f_i > 0$. The generalized SSA consists of two stages, the decomposition stage and the reconstruction stage, which are show in the following two subsection.

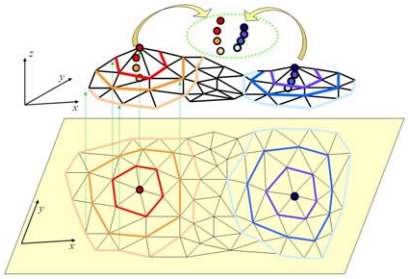


Figure 1. The mesh where height values are given to vertices of a graph on the plane.

2.2.1 Decomposition stage

The decomposition stage consists of the next two steps.

1st step: Embedding In the first step, let us define the linear operator $\boldsymbol{A}$ which maps F to matrix $\boldsymbol{X} = \boldsymbol{A}(F)$ as

$$\boldsymbol{A}(F) = \begin{pmatrix} FA_{0,0} & FA_{0,1} & \cdots & FA_{0,K-1} \\ FA_{1,0} & FA_{1,1} & \cdots & FA_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ FA_{L-1,0} & FA_{L-1,1} & \cdots & FA_{L-1,K-1} \end{pmatrix} \quad (1)$$

where

$$A_{l,k} = (a_{l,k,0},\ a_{l,k,1}, \cdots,\ a_{l,k,N-1})^{\mathrm{T}} \quad (2)$$

and the elements of $\boldsymbol{A}(F)$ are

$$FA_{l,k} = \sum_{n=0}^{N-1} a_{l,k,n} f_n. \quad (3)$$

We call the matrix (1) the trajectory matrix or the generalized trajectory matrix.

2nd step: Singular value decomposition In the second step, the singular value decomposition is applied to the trajectory matrix $\boldsymbol{X}$. Let $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^{\mathbf{T}}$. Denote by $\lambda_1, \ldots, \lambda_L$ the eigenvalues of $\boldsymbol{S}$ taken in the decreasing order of magnitude ($\lambda_1 \geq \ldots \geq \lambda_L \geq 0$), and by $U_1, \ldots, U_L$ the orthonormal system of the eigenvectors of the matrix $\boldsymbol{S}$ corresponding to these eigenvalues. Let $d = \max\{i \mid \lambda_i > 0\}$. We defines $V_i = \boldsymbol{X}^{\mathrm{T}} U_i / \sqrt{\lambda_i}$ and $\boldsymbol{X}^{(i)\mathrm{T}} = \sqrt{\lambda_i} U_i V_i^{\mathrm{T}}$ $(i = 1, \ldots, d)$. Then the singular value decomposition of the trajectory matrix $\boldsymbol{X}$ can be written as

$$\boldsymbol{X} = \boldsymbol{X}^{(1)} + \boldsymbol{X}^{(2)} + \ldots + \boldsymbol{X}^{(d)}. \quad (4)$$

The matrix $\boldsymbol{X}^{(i)}$ has rank 1. Therefore they are elementary matrices. The collection (λ_i, U_i, V_i) is called i-th eigentriple of singular value decomposition (4).

2.2.2 Reconstruction stage

3rd step: Reconstruction of the original series In the last step, each matrix in the decomposition (4) is transformed into a new series of length N. This step is called the reconstruction of the series.

The series $F^{(i)} = (f_0^{(i)}, f_1^{(i)}, \cdots, f_{N-1}^{(i)})$ is defined as the solution of the next optimization problem:

$$\min \sum_{i=1}^{d} \| \boldsymbol{X}^{(i)} - \boldsymbol{A}(F^{(i)}) \|^2 = \min \sum_{i=1}^{d} \sum_{l,k} \left(x_{l,k}^{(i)} - F^{(i)} A_{l,k}\right)^2 \quad (5)$$

$$\text{s.t.} \quad F = \sum_{i=1}^{d} F^{(i)}, \quad (6)$$

where the norm of the matrix is the Frobenius norm. If $A_{l,k} (0 \leq l \leq L-1, 0 \leq k \leq K-1)$ span N dimensional spaces, then the matrix $\sum_{l,k} A_{l,k} A_{l,k}^{\mathrm{T}}$ is regular, and consequently the solution of the expression (5) is given by

$$F^{(i)} = \left(\sum_{l,k} x_{l,k}^{(i)} A_{l,k}^{\mathrm{T}}\right) \left(\sum_{l,k} A_{l,k} A_{l,k}^{\mathrm{T}}\right)^{-1}. \quad (7)$$

Since this $F^{(i)}$ satisfies

$$\begin{aligned} \sum_{i=1}^{d} F^{(i)} \sum_{l,k} A_{l,k} A_{l,k}^{\mathrm{T}} &= \sum_{i=1}^{d} \sum_{l,k} x_{l,k}^{(i)} A_{l,k}^{\mathrm{T}} \\ &= \sum_{l,k} \left(\sum_{i=1}^{d} x_{l,k}^{(i)}\right) A_{l,k}^{\mathrm{T}} \\ &= \sum_{l,k} \left(FA_{l,k}\right) A_{l,k}^{\mathrm{T}} \\ &= F \sum_{l,k} A_{l,k} A_{l,k}^{\mathrm{T}}, \end{aligned}$$

the constraint (6) is satisfied automatically. The solution $F^{(i)}$ of the optimization problem (5) and (6) is obtained by the expression (7).

Finally, from the expression (7), the original series F is reconstructed as $F = \sum_{i=1}^{d} F^{(i)}$. These are the basic ideas of the generalized SSA.

2.3 Linear operator $\boldsymbol{A}$

In this subsection, we give a particular example of the linear operator $\boldsymbol{A}$ in the expression (1) that reflects the connectivity structure of the mesh.

Let P be a 3D polygonal mesh, and let v_k, $k = 0, 1, \cdots, N-1$ be the vertices of the mesh. Suppose that some scalar value f_k is assigned to each vertex

v_k, and let F be the series $F = (f_0, f_1, \cdots, f_{N-1})$. We define the distances $D_{v_k}(v_j)$ from v_k to v_j as the number of edges in the shortest path in the graph where the length 1 is given to the all edges, so-called the Dijkstra distance. Figure 2 shows an example of a part of the graph structure associated with the polygonal mesh. Let v_k be the vertex represented by the black dot in Figure 2. Then, the vertices v_j's with $D_{v_k}(v_j) = 1$ are as shown by empty circles, and the vertices with $D_{v_k}(v_j) = 2$ are as shown empty squares.

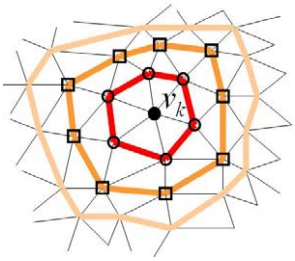


Figure 2. A vertex of the mesh and the set of vertices with the same Dijkstra distances.

Let the number of the rows of the trajectory matrix $\boldsymbol{A}(F)$ be $K := N$. Let the elements on the first row and the k-th column ($k = 0, 1, \cdots, N-1$) of the trajectory matrix $\boldsymbol{A}(F)$ be the value f_k given to the vertex v_k and let the elements on the l-th ($1 \leq l \leq L-1$) row and the k-th column be the average value of the values on the vertices whose Dijkstra distances from v_k are l. Therefore the k-th column of the matrix $\boldsymbol{A}(F)$ corresponds to the vertex v_k of the mesh. For example; for the vertex v_k in Figure 2. $(1, k)$ element of $\boldsymbol{A}(F)$ is f_k, $(2, k)$ element of $\boldsymbol{A}(F)$ is the average of the value f_j over the vertices represented by the empty circles, $(3, k)$ element of $\boldsymbol{A}(F)$ is the average of the value f_j over the vertices represented by the empty squares. Thus, let $FA_{l,k}$ be

$$FA_{l,k} = \frac{\sum_{D_{v_k}(v_j)=k} f_j}{\#\{D_{v_k}(v_j) = k\}} \tag{8}$$

where $\#\{D_{v_k}(v_j) = l\}$ is the number of the vertices whose Dijkstra distances from v_k are l. The linear operator $\boldsymbol{A}$ in the expression (8) is represented as

$$a_{l,k,n} = \begin{cases} \frac{1}{\#\{D_{v_k}(v_n)=l\}} & (D_{v_k}(v_n) = l), \\ 0 & (D_{v_k}(v_n) \neq l) \end{cases} \tag{9}$$

Note that, since the $A_{l,k}$ ($0 \leq l \leq L-1, 0 \leq k \leq K-1$) constructed as stated above span an N-dimensional space, the matrix $\sum_{l,k} A_{l,k} A_{l,k}^{\mathrm{T}}$ is usually regular. This is because $A_{1,k}$ ($0 \leq k \leq N-1$) is the vector where the k-th element is 1 and the other elements are 0.

The rows of $\boldsymbol{A}$ correspond to the vertices of the mesh, and the columns of $\boldsymbol{A}$ correspond to the set of vertices with the same Dijkstra distances. Here the linear operator reflects the connectivity structure of the mesh.

2.4 Laplacian trajectory matrix

We may not obtain sufficient amount of eigentriples for $\boldsymbol{A}(F)$ by our method in subsection 2.3. If the amount of eigentriples is small, some problems occur in engineering applications. For example, in the case of watermarking in section 4, we can not embed a lot of data. In order to solve this problem, we present a new Laplacian trajectory matrix in this subsection.

In subsection 2.3, f_h is assigned to each vertex v_h. In this subsection, in order to consider a new another trajectory matrix, let f'_h be Laplacian for a 3D polygonal mesh and f'_h is assigned to each vertex v_h. f'_h is defined as

$$f'_h = C\Delta f_h = C\left(\frac{\sum_{D_{v_h}(v_j)=1} f_j}{\#\{D_{v_h}(v_j) = 1\}} - f_h\right) \tag{10}$$

where C is a constant number. Figure 3 shows a mesh whose vertices v_h are transposed to Laplacian f'_h. Let F' be a series $F' = (f'_0, f'_1, \ldots, f'_{N-1})$. Since f'_h is a linear combination of elements of F, $A'_{i,j}$ exists such that $F'A_{i,j} = FA'_{i,j}$ and $\boldsymbol{A}(F')$ can be transposed to $\boldsymbol{A}'(F)$ as the following equation:

$$\begin{aligned} \boldsymbol{A}(F') &= \begin{pmatrix} F'A_{0,0} & \cdots & F'A_{0,K-1} \\ F'A_{1,0} & \cdots & F'A_{1,K-1} \\ \vdots & \ddots & \vdots \\ F'A_{L-1,0} & \cdots & F'A_{L-1,K-1} \end{pmatrix} \\ &= \begin{pmatrix} FA'_{0,0} & \cdots & FA'_{0,K-1} \\ FA'_{1,0} & \cdots & FA'_{1,K-1} \\ \vdots & \ddots & \vdots \\ FA'_{L-1,0} & \cdots & FA'_{L-1,K-1} \end{pmatrix} \\ &= \boldsymbol{A}'(F). \end{aligned} \tag{11}$$

A combined trajectory matrix $\boldsymbol{B}(F)$ is defined as

$$\boldsymbol{B}(F) = \begin{pmatrix} \boldsymbol{A}(F) \\ \boldsymbol{A}'(F) \end{pmatrix}. \tag{12}$$

If we use $\boldsymbol{B}(F)$, we can get the double sets of eigentriples for $\boldsymbol{A}(F)$.

We can expand $\boldsymbol{B}(F)$ likewise. Let $\boldsymbol{A}^0(F) = \boldsymbol{A}(F)$, $\boldsymbol{A}^1(F) = \boldsymbol{A}'(F)$ and $\boldsymbol{A}^i(F)$ be the duplicated Laplacian trajectory matrices by f^i_h, where f^i_h is defined as

$$f^i_h = C_i \Delta^i f_h. \tag{13}$$

$\boldsymbol{B}^i(F)$ is defined de by $\boldsymbol{A}^0(F), \cdots, \boldsymbol{A}^i(F)$ as

$$\boldsymbol{B}^i(F) = \begin{pmatrix} \boldsymbol{A}^0(F) \\ \hline \boldsymbol{A}^1(F) \\ \hline \vdots \\ \hline \boldsymbol{A}^i(F) \end{pmatrix}. \tag{14}$$

If we use $\boldsymbol{B}^i(F)$, we can get the i-fold sets of eigentriples for $\boldsymbol{A}(F)$.

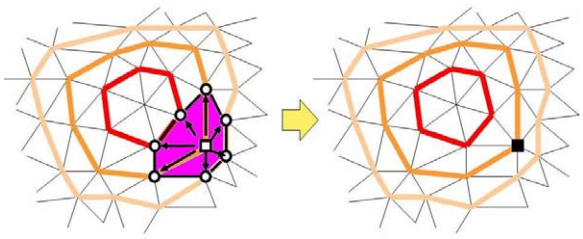


Figure 3. A mesh whose vertices are transposed to laplacian.

2.5 Comparisons of the decompositions

In this subsection, we compare the decomposition given by the generalized SSA, the basic SSA [6, 7, 8] and Laplacian matrix [15]. Table 1 shows their characteristics.

The 3D polygonal mesh represents the boundary of the three-dimensional region, and this boundary is a two-dimensional manifold. However, it is not easy to globally parameterize any two-dimensional manifold. Note that the three methods compared in this subsection do not require any parameterization of the boundary. On the other hand, the traditional multidimensional spectral decompositions such as the multidimensional Fourier transformation and the multidimensional wavelet transformation require the parameterization of the boundary, and hence cannot be used for general 3D polygonal meshes unless the mesh is partitioned into submeshes homeomorphic to disk. In this sense, these three methods are typical tools for the analysis of polygonal meshes.

Since the spectral decomposition of a mesh using the Laplacian matrix requires the eigenvalue decomposition of a matrix whose rank is the number of the vertices, the calculation cost is large and the method cannot be applied to huge meshes. On the other hand, our generalized SSA and the basic SSA require the eigenvalue decomposition of the matrix whose rank is determined by the linear operator e.g., the size of lag L. Moreover, we can choose a relative small value of L. Therefore, the calculation cost can be small, and consequently our generalized SSA and the basic SSA method can be applied to huge meshes.

The spectrum decomposition for 3D polygonal meshes is desired to be independent for the change of the vertex number. In other words, the basis of the spectrum decomposition is desired to be invariant from the change of the vertex number, because the shape of the mesh is invariant for the change of the vertex number. In the three methods, our generalized SSA and the method using the Laplacian matrix satisfy this requirement.

As stated above, our generalized SSA overcomes the respective demerits of the basic SSA and the method using the Laplacian matrix and can be a powerful new tool for the analysis of 3D polygonal meshes.

3 Spectral decomposition of 3D polygonal meshes using the generalized SSA

3.1 Spectral decomposition using the generalized SSA

In this subsection, we perform spectral decomposition of the 3D polygonal meshes using the generalized SSA.

Though we have been considering a scalar-value series $F = (f_0, f_1, \ldots, f_{N-1})$, we hereafter consider tri-value series $\boldsymbol{F} = (F_0, \ldots, F_{N-1})$ where $F_n = (f_{n,x}, f_{n,y}, f_{n,z})$ are the coordinates of the vertex v_n. Consequently, the trajectory matrix (1)is an $L \times 3K$ matrix

$$\boldsymbol{X} = \boldsymbol{A}(\boldsymbol{F}) = \begin{pmatrix} \boldsymbol{F}A_{1,1} & \cdots & \boldsymbol{F}A_{1,K} \\ \boldsymbol{F}A_{2,1} & \cdots & \boldsymbol{F}A_{2,K} \\ \vdots & \ddots & \vdots \\ \boldsymbol{F}A_{L,1} & \cdots & \boldsymbol{F}A_{L,K} \end{pmatrix}, \tag{15}$$

where

$$\boldsymbol{F}A_{l,k} = \left(\sum_{n=0}^{N-1} a_{l,k,n} f_{n,x}, \sum_{n=0}^{N-1} a_{l,k,n} f_{n,y}, \sum_{n=0}^{N-1} a_{l,k,n} f_{n,z} \right). \tag{16}$$

We perform singular value decomposition (SVD) for this trajectory matrices.

In our experiments, we used two popular mesh models, the bunny model (1494 vertices, 2915 faces) shown in Figure 4 (a). Figure 4 shows that the original bunny model mesh is decomposed using trajectory matrix with $L = 21$ and high frequency components are added gradually. Figure 4 (a) shows the original meshes. (b) shows the mesh constructed using the sum of the lowest frequency components. Figure 4 (c) or (d) shows the sum of 6 or 15 lower frequency components, respectively. Figure 5 is the decomposed mesh with $L = 10$ and Figure 6 is the decomposed mesh with $L = 5$. Figure 7 is the decomposed mesh of $L = 5$ using the

Table 1. Comparisons of three spectrum decomposition methods.

	decomposition using Laplacian matrix	decomposition using the basic SSA	decomposition using the generalized SSA
decomposition algorithm	eigenvalue decomposition of Laplacian matrix	the basic SSA	the generalized SSA
meaning of singular value or eigenvalue	frequency	power spectrum	power spectrum
parameterizations on 3D polygonal meshes	no parameterizations (merit)	no parameterizations (merit)	no parameterizations (merit)
spectrum decomposition and changes of the vertex number	independence (merit)	dependence (demerit)	independence (merit)
rank of the decomposed matrix (calculation cost)	order of the vertices of the mesh (large calculation cost: demerit)	order of the lag $L, (L < \frac{N}{2})$ (small calculation cost: merit)	order of the number of the rings (small calculation cost: merit)

mesh whose vertices are transposed to laplacian in subsection 2.4.

From Figures 4, 5, 6 and 7, we can confirm the following empirical fact. "Approximately, large singular values correspond to lower spatial frequencies, and small singular values correspond to higher spatial frequencies. Elementary matrices associated with higher singular values represent global shape features, while elementary matrices associated with lower singular values represent local or detail shape features. We made computational experiments in order to evaluate the performance of the proposed algorithms".

The computer used in this experiment is Precision 330 of Dell with Intel Pentium 4 2.8G Hz processor and 1GB memory. Programming language is Mathematica 4.0. Calculation times were 11 minite in case of bunny mode with $L = 10$.

4 Appliction — Watermarking 3D polygonal meshes

In this section, we propose a new method of watermarking for the 3D polygonal meshes.

4.1 What is watermarking ?

Digital watermarking is a technique for adding secret information called a watermark to various target objects data. A lot of papers on watermarking have been published [5]. However most of the previous researches have been concentrating on watermarking "classical" object data types, such as texts, 2D still images, 2D movies, and audio data. Recently, on the other hand, 3D objects data, such as 3D polygonal meshes and various 3D geometric CAD data, become more and more popular and important, and hence techniques to watermark 3D models also become more important [1, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20].

In the field of image watermarking, a majority of the watermarking algorithms published depends on some form of transformations, e.g., wavelet or Fourier transformations. This is because transformed domain techniques offer various advantages. For example, by modifying the spatial frequency band which human are not very sensitive to, we can make a watermark embedded in an image less visible. Moreover, the transformed domain is a suitable place to hide the secret data (watermarks). Therefore, in those techniques of watermarking, some kind of spectrum decomposition is required.

This section presents experiments and results of an algorithm that embeds watermarks into 3D polygonal meshes. The proposed method is based on a new kind of spectrum decomposition, and can be used for any mesh structures, for details refer to [6, 8, 9, 10]. We propose a new algorithm for embedding watermarks into 3D polygonal meshes based on the generalized SSA. The spectra of the 3D polygonal mesh are computed by the singular decomposition of the trajectory matrix, and the watermarks are embedded into the singular values. The watermark embedded by the algorithm is robust against similarity transformation (i.e., rotation, translation, and uniform scaling). It is also resistant against random noises added to vertex coordinates. Figure 8 is the outline of embedding a watermark.

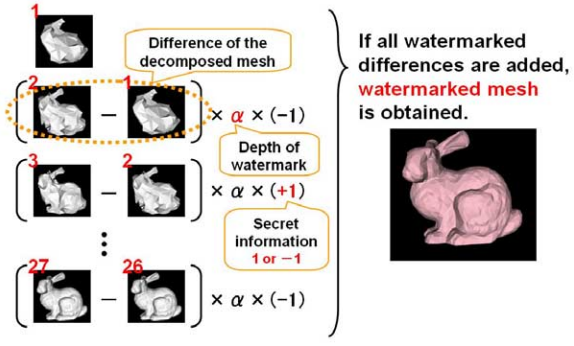


Figure 8. Outline of embedding watermark.

4.2 Experiments and results

4.2.1 Method

In our experiments, we used a popular mesh model, the bunny model (1494 vertices, 2915 faces) shown in Figure 4 (a).

We compare five watermarking methods. The first two watermarking methods are based on the

basic SSA using two kinds of the vertex series in [6, 8, 9, 10], which we call the "Euclidean norm method" and the "random order method". The third watermarking method is the Laplacian matrix method proposed by Ohbuchi et al. [15]. We call this watermarking method the "Ohbuchi's method". The forth watermarking method is based on our generalized SSA. We call this watermarking method the "generalized SSA ring 21" in the case of $L = 21$, and so on. The fifth watermarking method is based on our generalized SSA using Laplacian trajectory matrix in subsection 2.4. We call this watermarking method the "generalized SSA ring 5 Laplacian" in the case of $L = 5$. This method have 30 sets of eigentriples (i.e., $\times 3$ by xyz-coordinates and $\times 2$ by Laplacian).

In the"generalized SSA" method, we embedded 15 bits data, and each bit was embedded only once (i.e., chip rate is 1). In the other method, we embedded 15 bits data 20 times (i.e., chip rate is 20). If a mesh is fixed, a higher chip rate means a lower data capacity and higher robustness.

The watermark embedding amplitudes α is defined as $\alpha = \beta \times l$ where l is the largest length of the edges of the axis-aligned bounding box of the target mesh and β is defined as a ratio of the amplitude. In this experiments, $l = 156$ model was set. In Figure 9, the appearances for $\beta = 0.1, 1$ are presented. If α is larger, the watermark withstands against more disturbances, (for example, adding random noises and mesh smoothing) but the shape itself is distorted.

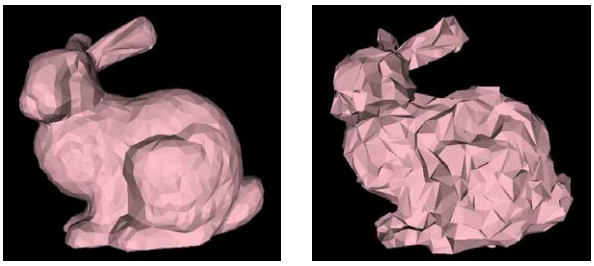
4.2.2 Appearances of watermarked meshes

Figure 9 show appearances of the watermarked meshes generated by the generalized SSA ring 21, while (a) and (b) show the watermarked meshes for $\beta = 0.1$ and 1, respectively. The appearances of (a) can hardly be distinguished from the appearances of the original mesh. Thus they are watermarked successfully. On the other hand, the appearances of the original meshes are not preserved in (b). Thus the watermarks are too large in those cases.

Table 2 shows RMS of the differences between the original meshes and the watermarked meshes devided by l. RMS (root mean square) is the mean of 2-norm between the vertices of the original mesh and the corresponding vertices of the watermarked mesh. In the appearances of the watermarked meshes, we cannot see much difference among the basic SSA methods and the generalized SSA method. In these experiments, we set $\beta = 0.1$ in the basic SSA and the generalized SSA, and $\beta = 0.0035$ in the Ohbuchi's method.

4.2.3 Robustness

We experimentally evaluated the robustness of our watermarks against the uniform random noises.



(a) $\beta = 0.1$ (b) $\beta = 1$

Figure 9. Watermarked bunny meshes.

Table 2. $\frac{\text{RMS}}{l}$ of original meshes and watermarked meshes.

Euclidean norm ($\beta = 0.1$)	0.0239
random order ($\beta = 0.1$)	0.0211
Ohbuchi's method ($\beta = 0.0035$)	0.0211
generalized SSA ring 21 ($\beta = 0.1$)	0.0228
generalized SSA ring 10 ($\beta = 0.1$)	0.0201
generalized SSA ring 5 Laplacian ($\beta = 0.1$)	0.0134

Uniform random noises Figure 10 shows the appearances of the watermarked mesh whose vertex coordinates were disturbed with uniform random noises with amplitude $\alpha \times \gamma$ ($\beta = 0.1$). Figure 10 (a) are the meshes with uniform random noises with $\gamma = 0.01$ and (b) are the meshes with uniform random noises with $\gamma = 0.1$. From Figure 10, we can see that the noises of $\gamma = 0.1$ deformed the appearances of the original meshes to a certain extent.

We counted the number of the bits reconstructed correctly; we repeated the experiment 100 times. The result is shown in Table 3. From this experiment, we can see that the watermark can withstand against uniform noises for $\gamma \leq 0.01$. Moreover, we cannot see much difference among the five methods.

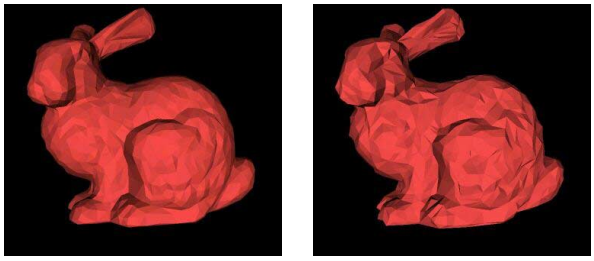
In the Euclidean norm method, the random order method and Ohbuchi's method, the same bit was embedded many times (20 times, for example) because each bit is very fragile. On the other hand, in the proposed method, each bit is embedded only once, but still the watermark can be reconstructed almost in the same accuracy as the other methods, as shown in Table 3. In this sense, the proposed watermark method is very robust against random noises.

This robustness is due to the characteristic of the linear operator $\boldsymbol{A}$. Since the elements of the generalized trajectory matrix are represented as the linear combinations of the vertices of the mesh, these linear combinations counteract the uniform random noises in this step. Therefore, since the generalized SSA counteracts the uniform random noises before spectrum decomposition, while the other methods counteract the uniform random

noises after spectrum decomposition; we can see almost the same robustness against random noises among these methods.

Table 3. Ratios of the correctly recovered watermarks under random noises.

	$\gamma = 0.1$	$\gamma = 0.01$
Euclidean norm	92%	100%
random order	98%	100%
Ohbuchi's method	98%	100%
generalized SSA ring 21	96%	100%
generalized SSA ring 10	99%	100%
generalized SSA ring 5 Laplacian	98%	100%



(a) $\gamma = 0.01$ (b) $\gamma = 0.1$

Figure 10. Bunny models to which uniform random noises with amplitude $\alpha \times \gamma$ ($\beta = 0.1$) are added.

5 Future Work

We have two future works. First future work is to develop new application area of generalized SSA. For that purpose, since there are large freedoms in the choice of the linear operator proposed in this paper, we need create new linear operator by considering the physical phenomenon of the target models. Second future work is to complete the theoretical framework of the generalized SSA.

Acknowledgement

This work is partly supported by the 21st Century COE Program on Information Science and Technology Strategic Core, and the Grant-in-Aid for Scientific Research (S) of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] O. Benedens. Geometry-based watermarking of 3D models. In *IEEE CG*, pp. 46–55, 1999.

[2] S. Broomhead and P. King. Extracting qualitative dynamics from experimental data. In *Physica D, Vol. 20*, pp. 217–2362, 1986.

[3] B. Elsner, J. and A. Tsonis, A. *Singular Spectrum Analysis - A New Tool in Time Series Analysis.* Plenum Press, 1996.

[4] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In *Proceedings of the Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6)*, pp. 296–307, 1998.

[5] K Matsui. *Basic of watermarks (in Japanese).* Morikita Shuppan Publishers, 1998.

[6] K. Murotani and K. Sugihara. Watermarking 3D polygonal meshes using the singular spectrum analysis. In *Proceedings of the 10th IMA International Conference on The Mathematics of Surfaces*, pp. 85–98, 2003.

[7] K. Murotani and K. Sugihara. Watermarking 3d polygonal meshes using the singular spectrum analysis. In *ISM Symposium Statistics, Combinatorics and Geometry*, pp. 20–22, 2003.

[8] K. Murotani and K. Sugihara. Generalized SSA and its applications to watermarking 3d polygonal meshes. In *METR METR 2004-17*, pp. 1–23, 2004.

[9] K. Murotani and K. Sugihara. Watermarking 3d polygonal meshes using generalized singular spectrum analysis. In *NICOGRAPH International Conference 2004 in Taiwan*, pp. 121–126, 2004.

[10] K. Murotani. Spectral decomposition method for three-dimensional shape models and its applications. In *Doctoral thesis (Information Science and Technology, University of Tokyo)*, 2004.

[11] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models. In *Proceedings of the ACM International Conference on Multimedia '97*, pp. 261–272, 1997.

[12] R. Ohbuchi, H. Masuda, and M. Aono. Targeting geometrical and non-geometrical components for data embedding in three-dimensional polygonal models. In *Computer Communications, Vol. 21*, pp. 1344–1354, 1998.

[13] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models through geometric and topological modifications. In *IEEE Journal on Selected Areas in Communication, Vol. 16, No. 4*, pp. 551–560, 1998.

[14] R. Ohbuchi, H. Masuda, and M. Aono. A shape-preserving data embedding algorithm for NURBS curves and surfaces. In *Proceedings of the Computer Graphics International'99*, pp. 7–11, 1999.

[15] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama. Watermarking 3D polygonal meshes in the mesh spectral domain. In *Proceedings of the Graphics Interface 2001*, pp. 9–17, 2001.

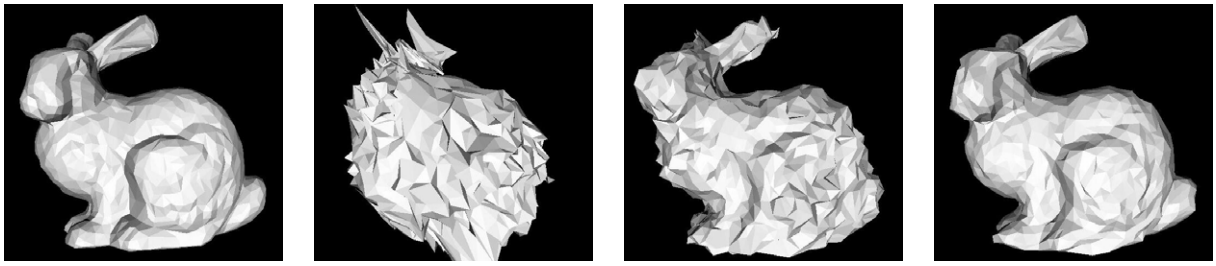
[16] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *ACM SIGGRAPH 1999*, pp. 69–76, 1999.

[17] R. Vautard, P. Yiou, and M. Ghil. Singular spectrum analysis: A toolkit for short noisy chaotic signals. In *Physica D, Vol. 58*, pp. 95–126, 1992.

[18] G. Wagner, M. Robust watermarking of polygonal meshes. In *Proceedings of Geometric Modeling & Processing 2000*, pp. 201–208, 2000.

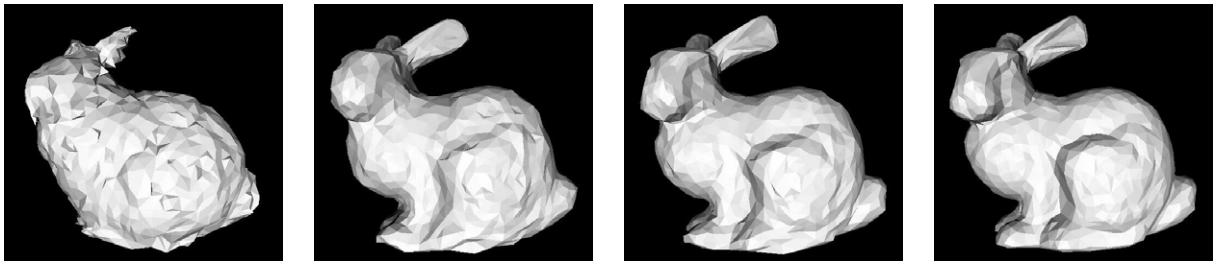
[19] B-L. Yeo and M. Yeung, M. Watermarking 3D objects for verification. In *IEEE CG&A*, pp. 36–45, 1999.

[20] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. In *Computers & Graphics, Vol. 25*, pp. 409–420, 2001.



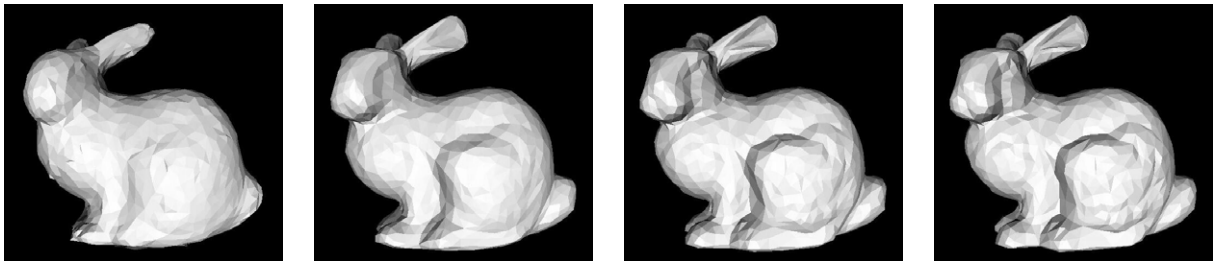
(a) Original mesh (b) $\hat{d} = 1$ (c) $\hat{d} = 6$ (d) $\hat{d} = 15$

Figure 4. The decomposed bunny model of $L = 21$. The sum of $\hat{d}$ lower frequency components.



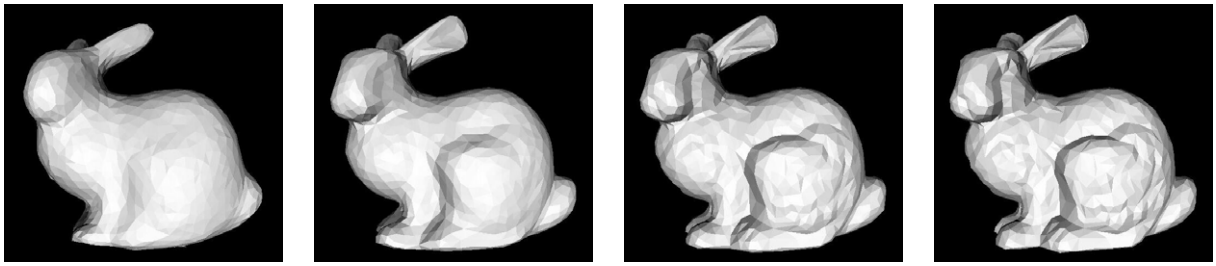
(a) $\hat{d} = 1$ (b) $\hat{d} = 3$ (c) $\hat{d} = 5$ (d) $\hat{d} = 8$

Figure 5. The decomposed bunny model of $L = 10$. The sum of $\hat{d}$ lower frequency components.



(a) $\hat{d} = 1$ (b) $\hat{d} = 2$ (c) $\hat{d} = 3$ (d) $\hat{d} = 4$

Figure 6. The decomposed bunny model of $L = 5$. The sum of $\hat{d}$ lower frequency components.



(a) $\hat{d} = 1$ (b) $\hat{d} = 2$ (c) $\hat{d} = 3$ (d) $\hat{d} = 9$

Figure 7. The decomposed bunny model of $L = 5$ using the mesh whose vertices are transposed to laplacian. The sum of $\hat{d}$ lower frequency components.