Journal of Computational Design and Engineering, Vol. 1, No. 2 (2014) 96~102
www.jcde.org

Direct construction of a four-dimensional mesh model from a three-dimensional object with continuous rigid body movement

Ikuru Otomo*, Masahiko Onosato and Fumiki Tanaka
Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido, Japan

(Manuscript Received September 12, 2013; Revised October 26, 2013; Accepted November 1, 2013)

Abstract

In the field of design and manufacturing, there are many problems with managing dynamic states of three-dimensional (3D) objects. In order to solve these problems, the four-dimensional (4D) mesh model and its modeling system have been proposed. The 4D mesh model is defined as a 4D object model that is bounded by tetrahedral cells, and can represent spatio-temporal changes of a 3D object continuously. The 4D mesh model helps to solve dynamic problems of 3D models as geometric problems. However, the construction of the 4D mesh model is limited on the time-series 3D voxel data based method. This method is memory-hogging and requires much computing time. In this research, we propose a new method of constructing the 4D mesh model that derives from the 3D mesh model with continuous rigid body movement. This method is realized by making a swept shape of a 3D mesh model in the fourth dimension and its tetrahedralization. Here, the rigid body movement is a screwed movement, which is a combination of translational and rotational movement.

Keywords: Four-dimensional mesh model; Three-dimensional mesh model; Fourth dimension; Rigid body movement

1. Introduction

In the field of design and manufacturing, there are various applications that represent the time change of a three-dimensional (3D) object. They are realized by preparing time-series data of 3D objects, giving movement operation to an initial state, or by interpolating among different states as shown in Figures 1(a), 1(b) and 1(c). However, these methods are insufficient to represent spatio-temporal changes of 3D objects' shapes, positions and orientations continuously and with high accuracy. Therefore, the four-dimensional (4D) mesh model and its modeling system have been proposed [11]. The four-dimensional model enables the representation of dynamic states of 3D objects as a static geometric entity in a higher-order space of space-time extension as presented in Figure 1(d). For example, it helps to solve problems of collision avoidance between 3D objects including deformation and motion. Another example is representing the continuous process of workpiece transformation in five-axis machining using the 4D mesh model [6]. However, the construction of a 4D mesh model is realized with a time-series 3D voxel data based method that is memory-hogging and requires longer computing time.

*Corresponding author. Tel.: +81-11-706-6438, Fax.: +81-11-706-6438
E-mail address: otomo@dse.ssi.ist.hokudai.ac.jp

doi: 10.7315/JCDE.2014.010

The objective of this study is to propose another approach of constructing a 4D mesh model using a 3D mesh model with rigid body movement. This method enables construction of a 4D mesh model by decreasing the number of vertices, tetrahedra and creation-time more than in the existing method.

At first, this paper describes the related works and the overview of the 4D mesh model and its modeling system. Then, the proposed method of constructing the 4D mesh model is explained in detail. Finally, a comparison with the time-series voxel data based method is reported.

2. Related studies and our previous works

In general, the representation of the time change of a 3D object is realized by preparing time-series data of 3D objects,

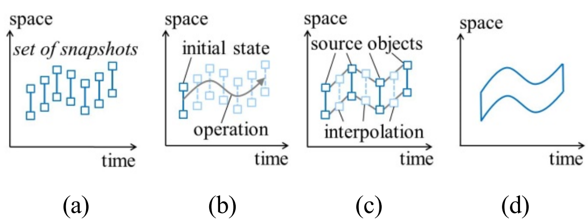


Figure 1. Several approaches of representing time change of a 3D object (space is reduced to 1D): (a) time-series data, (b) initial state and operation of time change, (c) interpolation, (d) 4D model.

giving movement operation to an initial state, or by interpolating among different states. An example of preparing time-series data of 3D objects is the 3D Video [9]. 3D Video is a medium that records a real time-varying 3D object using several cameras from several perspectives. The object's 3D shape is reconstructed using a volume intersection method, and finally, the constructed 3D models are colored by applying the real camera images.

Several physics simulations and machine tool simulations are realized by giving movement operation to an initial state. The state is updated in sequence for each unit time. In the physics simulations, the behavior of a 3D object is determined by the given "force" that is the factor of time change. Recently, physics simulation can be implemented easily into a computer using physics engines like PhysX [13] and Havok [4]. As well as in the machine tool simulations, the movement of a machine tool is driven by operation commands like G-code [5].

An example of interpolating among different states is morphing of a 3D polygon model [7]. Morphing is a technique to change a source object gradually through intermediate objects into a target object. It is realized by constructing the correspondence map among the meshes of the source object and the target object, and computing interpolating points between both objects.

As explained above, the current approaches of representing the time change of a 3D object are realized by preparing a set of static states. They only have discrete information along the time axis and cannot describe the time change continuously with high accuracy. Furthermore, since the information along the time axis is discrete, problems can occur on collision detection among several objects. Therefore, we proposed the 4D mesh modeling and its modeling system. In our previous works [6, 11, 12], a 4D mesh model is constructed from time-series 3D voxel data. Voxel is a simple way to represent 3D objects' shapes and it is also possible to apply higher dimensional Marching Cubes algorithm to construct a 4D mesh model. However, dealing with high-resolution voxel models consumes a large amount of memory.

3. Overview of four-dimensional geometric modeling

3.1 Four-dimensional mesh model

In this study, the fourth dimension (4D) is a 4D Euclid space that includes three-dimensional space and one-dimensional time. Figure 2 shows the four topological elements of 4D topologies: vertices, edges, faces and cells. A cell is a 3D subspace in a 4D space and the simplest cell is a tetrahedron. A pentachoron is a 4D simplex that is bounded by five tetrahedral cells. In addition, a 4D cube is called a hypercube and consists of 16 vertices, 32 edges, 24 faces and 8 cells. A 4D mesh model is defined as the 4D object model that is bounded by tetrahedral cells. As shown in Figure 3, 3D models are obtained by extracting the cross-section of a 4D mesh model.

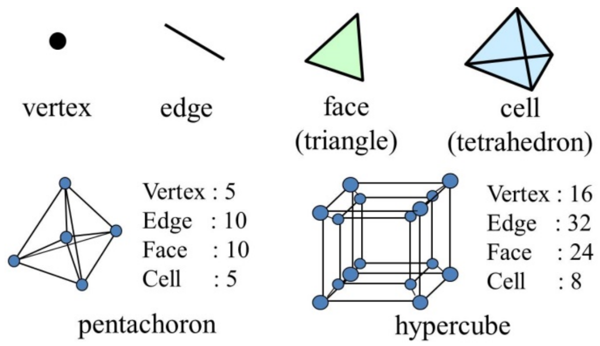


Figure 2. Four elements of 4D topologies and the basic 4D shapes.

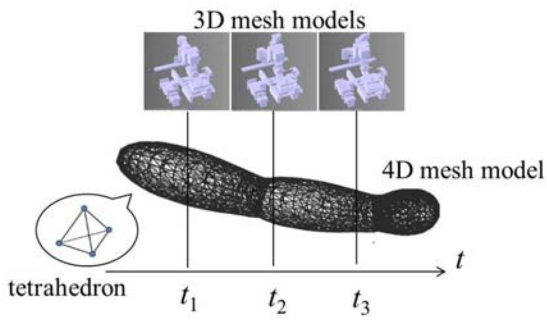


Figure 3. 4D mesh model representation.

3.2 4D mesh modeling system

Figure 4 describes the flow of the current 4D mesh modeling system. Time-series 3D voxel data are needed to construct a 4D mesh model (Figure 4 A), and for weaving from the 3D voxel models to a 4D mesh model, the 4D Marching Cubes algorithm [12] is used (B and C). The 4D Marching Cubes algorithm is an extension of the Marching Cubes algorithm [8] to 4D, and it is based on a higher dimensional isosurfacing algorithm [3].

There are various applications to manage the 4D mesh model (D, E, F and G). It is possible to project a 4D mesh model into 3D and visualize it as a 3D envelope model (D). The 3D envelope model is constructed by applying the Ball-Pivoting algorithm [2] to the 4D mesh model's vertices that is projected into the 3D space. In addition, cross-section extraction from the 4D mesh model by hyper-plane cutting is also possible (E). It is realized by applying the Marching Tetrahedra algorithm [10] to each tetrahedral cell of the 4D mesh model. Employing the collision detection method between 4D objects (F), a method of collision resolution by shape deformation for collision avoidance between moving objects is also proposed (G). The 4D collision detection method is an extension of collision detection method using AABB tree in 3D [17].

As mentioned above, the construction of a 4D mesh model

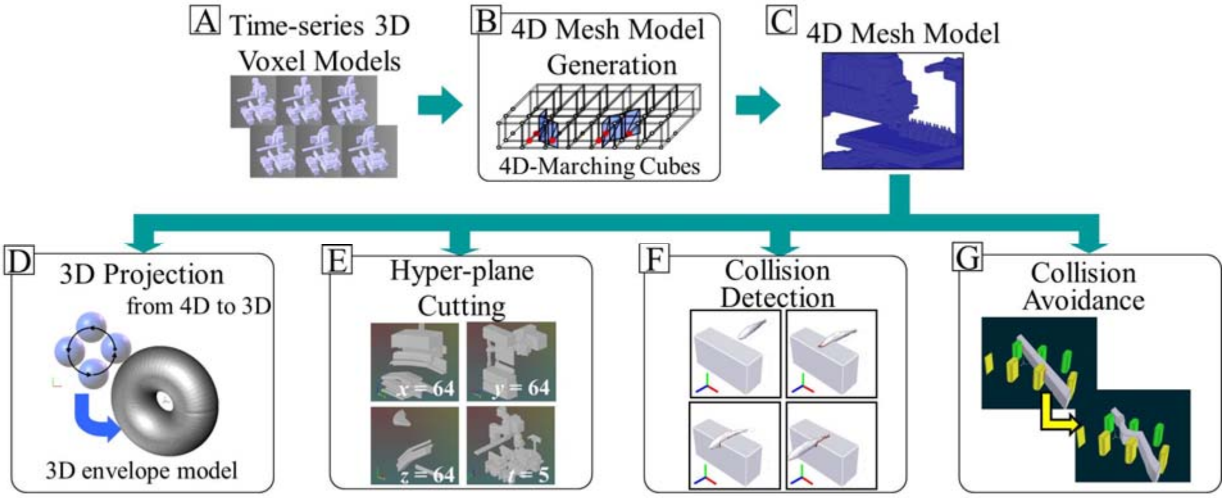


Figure 4. Flow of the 4D mesh modeling system.

is currently realized with time-series 3D voxel data based method. We propose another approach for constructing a 4D mesh model from a 3D mesh model with rigid body movement.

4. Construction of a 4D mesh model from 3D mesh model with rigid body movement

4.1 Tetrahedralization of the swept shape of 3D mesh model

In this research, from a 3D mesh model and parameters of translational and rotational movements, a swept shape of the 3D mesh model along the spatio-temporal axes is obtained. This obtained swept shape in 4D is tetrahedralized to represent it as a 4D mesh model. Here, we do not consider a movement that moves backward through time. In other words, every time axis element in translational movement vectors must be positive. The proposed method mainly consists of following four steps:

- Step 1. Tetrahedralize the inside of the 3D mesh model at the initial state;
- Step 2. Give movements to the 3D mesh model in sequence and tetrahedralize the 4D swept shape repeatedly;
- Step 3. Tetrahedralize the inside of the 3D mesh model at the final state;
- Step 4. Construct a 4D mesh model by merging tetrahedra generated in Step 1, Step 2 and Step 3.

Step 1 and Step 3 are necessary processes to represent both ends normal to *t*-axis of a 4D mesh model also using a set of tetrahedra. Therefore, it is needed to tetrahedralize the inside of a 3D mesh model. In this regard, it is necessary to add new vertices since it is known that some polyhedra have no tetrahedralization at all [1, 14]. Furthermore, additional vertices should not be added on the boundary shapes and also the boundary shapes should be preserved. Thus, the Constrained

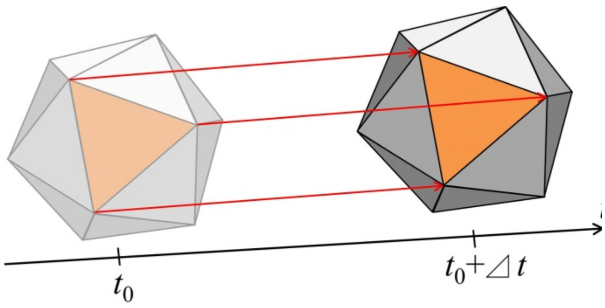


Figure 5. Swept shape of a triangular mesh along the spatio-temporal axes.

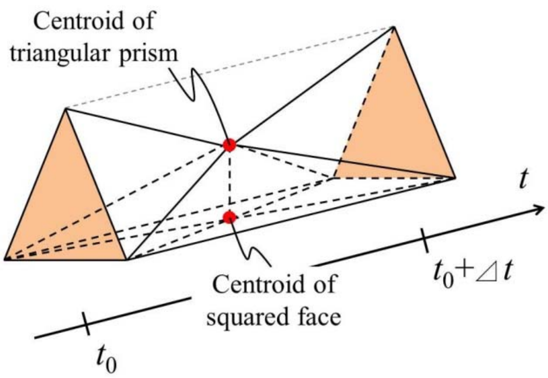


Figure 6. Tetrahedralization of a triangular prism.

Delaunay Tetrahedralization method [15] is used. It considers the boundary shapes as constrained conditions and executes the tetrahedralization. On executing the tetrahedralization of a 3D mesh model, it has to be confirmed that there is no lack or interference of meshes in the 3D mesh model. Otherwise, the inside/outside decision of the 3D mesh model may fail and the tetrahedralization may not finish correctly. In our software program, the tetrahedralization method for a 3D mesh model is implemented by TetGen [16].

Concerning Step 2, the swept shape of a triangular mesh along the spatio-temporal axes is a triangular prism (Figure 5). To tetrahedralize each triangular prism, four new vertices are added: one vertex at the centroid of triangular prism and three vertices at each centroid of squared faces (Figure 6). Two tetrahedra are generated from triangular faces and the centroid of the triangular prism, and four tetrahedra, including the centroid of the triangular prism and the centroid of squared face as elements, are generated from each squared face. Finally, a triangular prism is divided into 14 tetrahedra.

In this study, since the correspondence relationship between the vertices of before and after the movement has to be obvious, we cover only rigid body movement. Therefore, more work should be carried out on deformation in the future.

4.2 Deciding step size of a rotation angle using volume change as index

On applying the proposed method to rotational movement, a “twisted” triangular prism is generated by joining corresponding vertices of before and after the rotation (at time $t = t_0$ and $t = t_0 + \Delta t$) (Figure 7). Therefore, a rotational angle has to be divided in such a way that the distortion at $t = t_0 + \Delta t / 2$ becomes smaller than a given threshold.

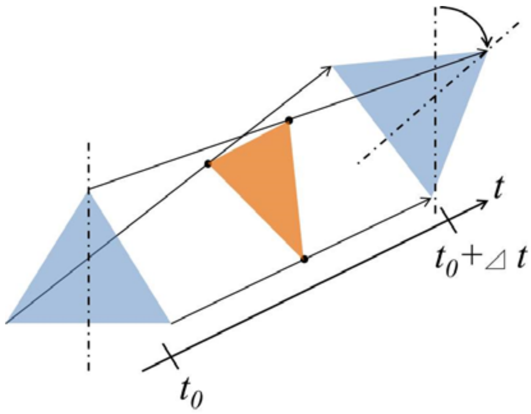


Figure 7. Twisted triangular prism.

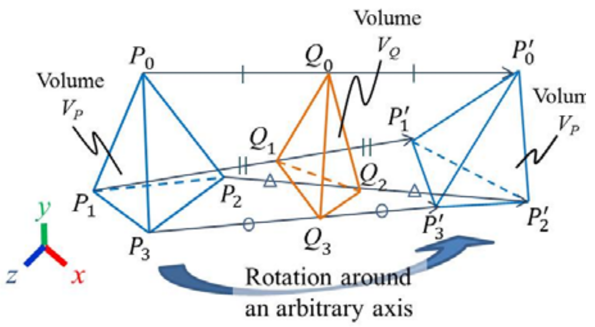


Figure 8. Volume change of a tetrahedron caused by a rotation.

As illustrated in Figure 8, in the 3D space, consider a tetrahedron composed of four vertices $P_i=[x_{P_i}, y_{P_i}, z_{P_i}]^T$ (i = 1, 2, 3, 4) whose volume is V_P, and the tetrahedron rotates θ around an arbitrary axis that passes an arbitrary point. We assume that P_i' is the four points after the rotation and each center between P_i and P_i' is Q_i. In addition, the volume of the tetrahedron composed of vertices Q_i is defined as V_Q.

If the ratio V_Q/V_P is defined as the volume ratio R_V, it is defined as Eq. (1) and its graph is shown in Figure 9.

$$R_V = \frac{1+\cos\theta}{2} \tag{1}$$

Since $|\cos\theta| \le 1$, $0 \le R_V \le 1$ holds. As the inside of a 3D mesh model can be represented by a set of tetrahedra, Eq. (1) holds on a 3D mesh model too.

If an acceptable lower limit of volume change $R_{V_{\min}}$ (where $0 \le R_{V_{\min}} < 1$) is given by users, the step size of a rotation θ_{step} (where $0° < \theta_{\text{step}} \le 180°$) is given by Eq. (2).

$$\theta_{\text{step}} = \cos^{-1}\left(2R_{V_{\min}} - 1\right) \tag{2}$$

Figure 10 shows the relationship between $R_{V_{\min}}$ and θ_{step}. For example, if $R_{V_{\min}}$ = 0.998 then θ_{step}= 5.1. By dividing a rotational angle θ by θ_{step}, the volume change of a 3D mesh model fits into the acceptable range.

5. Results of constructing 4D mesh models

5.1 Comparison with the time-series 3D voxel data based method

By preparing time-series 3D voxel data and a 3D mesh model, two 4D mesh models are constructed. The number of space partitions of the time-series 3D voxel data is 128×128×128, and the number of time steps is 64.

In the time-series 3D voxel data, a cube whose length is 20 voxels moves 1 voxel to the +*z*-axis direction for each time step. In the 3D mesh model, a cube is prepared whose length is 20. The cube is represented by 14 vertices and 24 triangular meshes. It moves +1.0 to the *z*-axis direction for each time

Table 1. Data of constructed 4D mesh models.

	Time-series 3D voxel data based method	3D mesh model based method
Number of vertices	197,828	4,750
Number of tetrahedra	1,185,280	21,552
Construction time[sec]	4.63	0.06

* OS:Windows XP, CPU:AMD Opteron 2224SE 3.2GHz, RAM:64GB

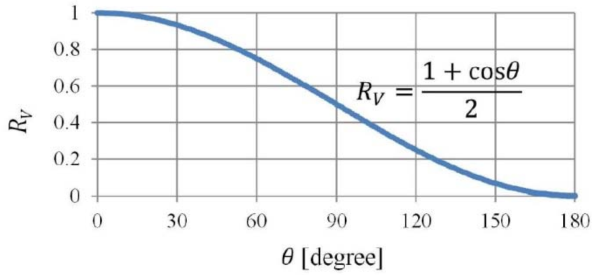


Figure 9. Relationship between the volume ratio R_V and the rotational angle θ.

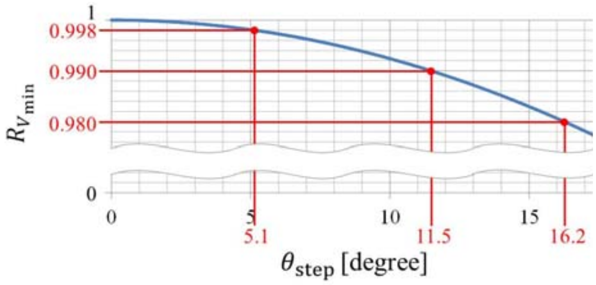


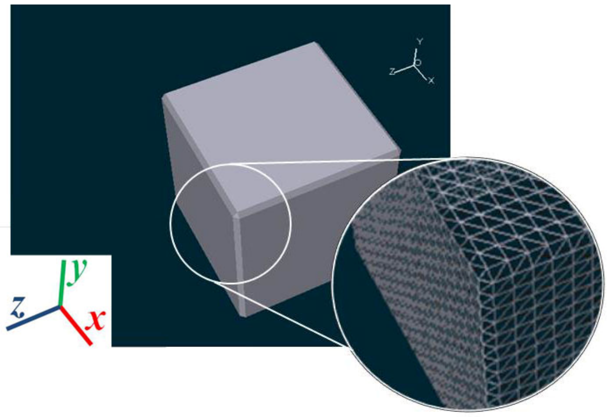
Figure 10. Relationship between the acceptable lower limit of volume change $R_{V_{\min}}$ and the step size of a rotation θ_{step}.

interval 1.0 during $t = 0 \sim 64$ in the virtual field.

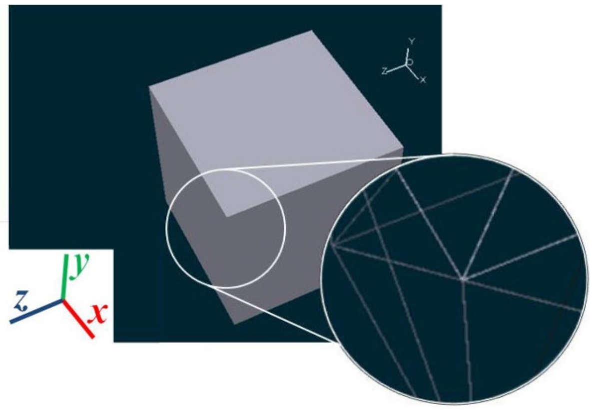
In Figure 11, the 3D mesh models that are extracted from a slicing hyper plane perpendicular to t-axis are shown. It represents the state of the 3D model at a specified time. Since the time-series 3D voxel data based method is realized using the 4D Marching Cubes algorithm, it cannot represent sharp edges and corners as shown in Figure 11(a). On the other hand, as presented in Figure 11(b), such a problem does not occur with the 3D mesh model based method. Table 1 shows the number of vertices, the number of tetrahedra, and the processing time for the construction of each 4D mesh model. As a result, the proposed method can construct a 4D mesh model more quickly that has less vertices and tetrahedra than that of the existing method.

5.2 Construction of a 4D mesh model from a 3D mesh model with rotation

In preparing a 3D mesh model of a squirrel (Figure 12) as an input, a 4D mesh model is constructed. The given movement is a screwed movement that rotates 360 degrees around the x-axis and translates +20.0 to the x-axis direction simultaneously during the time $t = 0 \sim 100$ in the virtual field. The acceptable lower limit of volume change $R_{V_{\min}}$ is 0.9976, and as a result, the movement is divided into 65 steps (5.538 degrees of rotation for each). The processing time for the construction of the 4D mesh model is shown in Figure 13. The 4D mesh model has 3,909,554 vertices and 18,266,982 tetrahedra. In Figures 14 and 15, the 3D mesh models that are extracted from the slicing hyper plane perpendicular to x, y, z, or t-axis are shown. From these figures, we can verify that a 3D object with movement is represented as a 4D model.



(a)



(b)

Figure 11. Results of cross-section extraction to 4D mesh models perpendicular to t-axis (t = 32): (a) time-series 3D voxel data based method, (b) 3D mesh model based method.

6. Conclusions

In this study, a new approach of constructing a 4D mesh model directly from a 3D mesh model with rigid body movement is proposed. The main contributions of this report are as follows:

- The method of constructing a 4D mesh model directly from a 3D mesh model with rigid body movement was explained in detail. We proposed a tetrahedralization method of 4D swept shapes from triangular meshes that represented rigid body movement and time change of a 3D mesh model. At this time, the swept shape of a triangular mesh was divided into 14 tetrahedra. In addition, to make the constructed 4D mesh model a closed shape, we proposed a method of tetrahedralizing the inside of the 3D mesh model and adding the generated tetrahedra to the 4D mesh model.

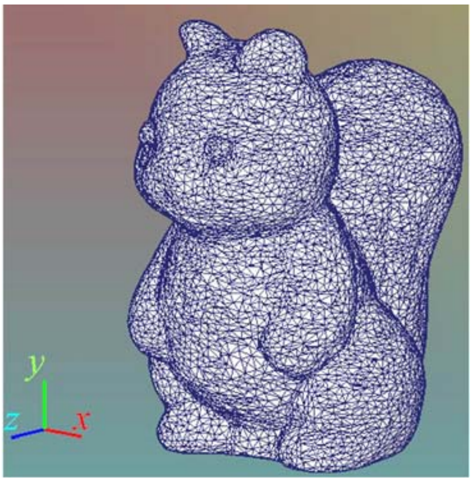


Figure 12. 3D mesh model of a squirrel.

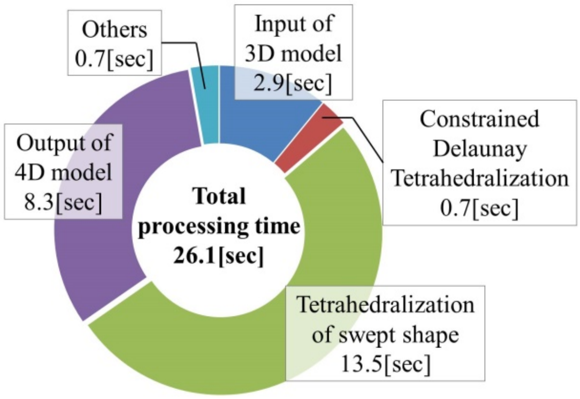
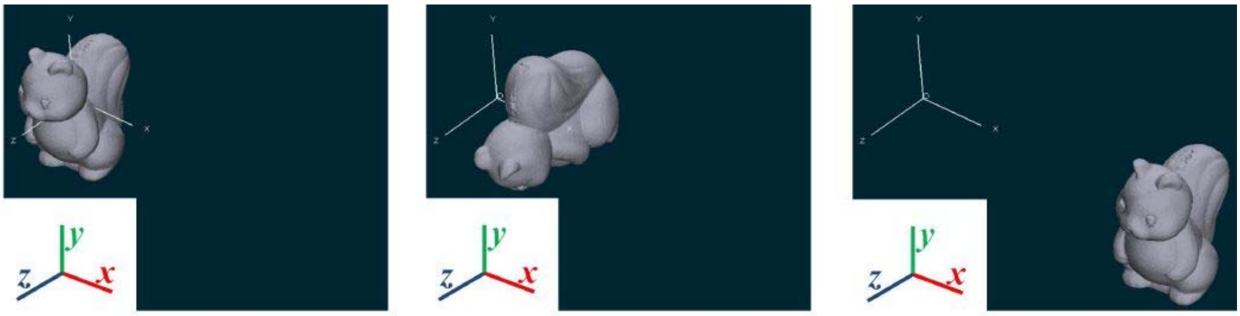
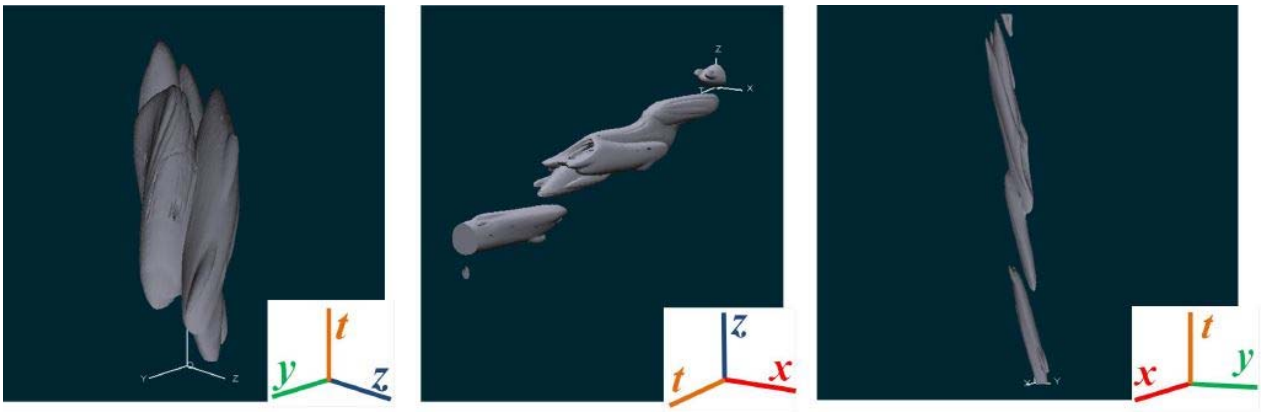


Figure 13. Processing time.



(a) (b) (c)

Figure 14. Results of cross-section extraction to 4D mesh models perpendicular to t-axis: (a) t = 1.0, (b) t = 20.0, (c) t = 100.0.



(a) (b) (c)

Figure 15. Results of cross-section extraction to 4D mesh models perpendicular to x, y, or z-axis: (a) x = 3.0, (b) y = 3.3, (c) z = 3.5.

- By analyzing the volume change that occurred when joining corresponding vertices of a 3D mesh model before and after a rotational movement, we proposed a method of deciding the step size of a rotational angle in such a way that the volume change of a 3D mesh model fitted into the acceptable range.
- It was shown that the proposed method enabled significant reduction of computing time and data volume of the constructed 4D mesh model as compared to the time-series 3D voxel data based method. This fact was shown with an example.

The remaining issues of this study are to evaluate the quality of tetrahedral meshes of constructed 4D mesh model, and to apply the proposed method to deformation. Speeding up the proposed method using GPU is also considered.

Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Number 24656106.

References

[1] Bagemihl F. On indecomposable polyhedra. The American Mathematical Monthly. 1948; 55(7): 411-413.

[2] Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G. The ball-pivoting algorithm for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics. 1999; 5(4): 349-359.

[3] Bhaniramka P, Wenger R, Crawfis R. Isosurfacing in higher dimensions. In: Proceedings of the 11th annual IEEE Visualization conference; 2000 Oct 8-13; Salt Lake City, UT; p. 267-273.

[4] Havok [Internet]. Dublin (Ireland): Telekinesys Research Ltd.; c1999-2013 [cited 2013 Nov 25]. Available from: http://havok.com/

[5] International organization for standardization. ISO 6983-1: Numerical control of machines -Program format and definition of address words - Part 1: Data format for positioning, line motion and contouring control. Geneva (Switzerland): The Organization; 1982.

[6] Kameyama H, Otomo I, Onosato M, Tanaka F. Representing continuous process of workpiece transformation in five-axis machining using spatio-temporal model. In: Proceedings of Asian Conference on Design and Digital Engineering; 2012 Dec 6-8; Niseko, Japan; Accompanied by: 1 USB memory stick.

[7] Lee AWF, Dobkin D, Sweldens W, Schröder P. Multiresolution mesh morphing. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques; 1999 Aug 8-13; Los Angeles, CA; p. 343-350.

[8] Lorensen WE, Cline HE. Marching cubes: A high resolution 3D surface construction algorithm. ACM SIGGRAPH Computer Graphics. 1987; 21(4): 163-169.

[9] Moezzi S, Tai LC, Gerard P. Virtual view generation for 3D digital video. IEEE Multimedia. 1997; 4(1): 18-26.

[10] Müller H, Wehle M. Visualization of implicit surfaces using adaptive tetrahedrizations. In: Scientific Visualization Conference; 1997 Jun 9-13; Dagstuhl, Germany; p. 243-250.

[11] Onosato M, Kawagishi R, Kato K, Date H, Tanaka F. Four-dimensional mesh modeling for spatio-temporal object representation. In: Proceedings of Asian Conference on Design and Digital Engineering; 2010 Aug 25-28; Jeju, Korea; p. 579-589.

[12] Onosato M, Saito Y, Tanaka F, Kawagishi R. Weaving a four-dimensional mesh model from a series of three-dimensional voxel models. In: Proceedings of 10th Annual International CAD Conference and Exhibition; 2013 Jun 17-20; Bergamo, Italy; Accompanied by: 1 USB memory stick.

[13] PhysX [Internet]. Santa Clara (CA): Nvidia Corporation; c2013 [cited 2013 Nov 25]. Available from: https://developer.nvidia.com/technologies/physx

[14] Schönhardt E. Über die Zerlegung von Dreieckspolyedern in Tetraeder [About the subdivision of triangular polyhedron into tetrahedra]. Mathematische Annalen. German. 1928; 98(1): 309-312.

[15] Shewchuk JR. Constrained delaunay tetrahedralizations and provably good boundary recovery. In: Proceedings of the 11th International Meshing Roundtable; 2002 Sep 15-18; Ithaca, NY; p. 193-204.

[16] Si H. TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator [Internet]. Berlin (Germany): Weierstrass Institute for Applied Analysis and Stochastics, Research group of Numerical Mathematics and Scientific Computing; [cited 2013 Nov 25]. Available from: http://tetgen.org

[17] van den Bergen G. Efficient collision detection of complex deformable models using AABB trees. Journal of Graphics Tools. 1997; 2(4): 1-13.