Annotation and View Synchronization of Shared 3D Models

Hiroshi Shimada, Kengo Imae
Graduate School of Frontier Informatics
Kyoto Sangyo University,
Kyoto, Japan
Email: {i1658083,i1558048}@cc.kyoto-su.ac.jp

Naohiro Hayashibara
Faculty of Computer Science and Engineering
Kyoto Sangyo University,
Kyoto, Japan
Email: naohaya@cc.kyoto-su.ac.jp

***Abstract*—Many pictures and videos have been shared on the Web. Thus, it is easy to find what one needs by using a Web search engine. Given the popularity of 3D printers, it is expected that 3D models will also be shared in the same manner. We focus here on 3D models with camera viewpoint and propose a prototype implementation of the annotation sharing system of 3D models. We also implement the synchronization mechanism for 3D models based on Publish/Subscribe model. Then, we evaluate the resource overhead of a server and the response time of clients in our prototype.**

***Keywords*–3D model; annotation; viewpoint synchronization; publish/subscribe systems**

I. Introduction

Pictures and videos are major contents on the Web. We can find favorite ones using a search engine and share them with social network services. Sharing annotations on such contents is particularly interesting in terms of promoting communication.

On the other hand, 3D models are getting popular in the area of virtual reality and 3D printing in recent years. So, they are expected to be shared on the Web like pictures and videos, and will be used in teleconference and distance education for an intuitive awareness (e.g., [1]). However 3D models have a problem that the appearance changes by rotation and zoom operations. For this reason, we have to consider a new way of annotation on 3D models and their sharing.

Now, we take a look at the difference between 2D contents (e.g., pictures and movies) and 3D contents in terms of annotation.

2-Dimensional Contents: One of the features of 2-dimensional (2D) contents is to be able to display all information in the same plane. In the case of pictures, annotations can be added simply on pictures. Video contents include much more information than pictures. A video can be considered to be a sequence of pictures, and changes the view over time. However, it is still the same as a picture at the moment. Thus, annotations for videos is also realized by adding it directly on the contents.

3-Dimensional Contents: For shared 3-dimensional (3D) contents, it is important to consider a view through a camera. When a user operates a shared 3D model, the camera position is changed by the operation. Thus, the view changes according to the position. To represent the intention of an annotation precisely, it should be associated with the camera position. It is also appeared with the same view as when it was added.

In this paper, we focus on the problem on the visual appearance of a 3D model caused by operations, such as rotation and zoom. When a user adds some annotation on a 3D model, it should be stored together with the current camera position. To display annotations, it is important to select some of them according to the camera to emphasize their intention. Despite the problem, most of the existing approaches display all annotations of a 3D model in the same plane.

Some of the recent research work take into account of the visual appearance of a 3D model and annotations are associated with a part of the model by using explicit links. However, such approaches become more complicated when many annotations are added by users.

Contribution: We propose (i) the approach of the perspective-oriented display of annotations and (ii) the viewpoint synchronization mechanism for 3D models. The former one is realized by annotations with position information of a camera. Each annotation is associated with an implicit link from the camera position, which realizes the user's perspective, to the corresponding 3D model going through the annotation. The latter one is the viewpoint sharing mechanism based on the Publish/Subscribe (Pub/Sub) model. It realizes the synchronization on the visual appearance for 3D models among multiple users. Moreover, we have evaluated the proposed system regarding resource consumption and the response time. Our contribution is expected to be useful for distance education and teleconferencing using 3D models.

II. Related work

In this section, we introduce several existing work related to mechanisms to annotate 3D models and applications using 3D models.

A. Annotation on shared 3D models

Although there are a lot of work on the annotation of 3D models such as ones in the context of CAD drawing [2], [3], we now explain some of them which have motivation similar to ours.

Kahan et al. developed a Web-based shared annotation system for Web documents, called Annotea [4]. It is based on an open Resource Document Framework (RDF) infrastructure standardized by W3C. Annotations, which

 ISBN: 978-1-61208-538-8

are attached by users, are managed by Annotation servers separated from the corresponding document. The protocol between clients and the server has been defined as Annotea Protocols [5].

Kadobayashi et al. proposed notions of Physical viewpoint for annotations of 3D models [6]. It is the way to put annotations based on the camera position at which a user is looking. Hunter and Yu proposed the 3D semantic annotation system for digital 3D artifacts [7] and crystallography models [8] including annotations. The problem of those systems is to be complicated if many annotations (i.e., windows and links) have been put into a 3D model.

1) Teleconference and e-Learning using 3D models: There are so many applications using 3D models for realizing the virtual environment in teleconferencing, e-Learning, and training in the medical, educational and business fields. In particular, they are sophisticated in terms of devices because lots of VR devices such as head-mount display (HMD) have been developed in recent years.

Goeser et al. developed the web-based 3D virtual framework for e-Learning, called VIEW [9]. It consists of two modules for tensile testing and mechanical assembling. In the result of their experiments, the proposed system and modules using 3D models give students the learning experience equivalent to the physical one.

Kleven et al. developed a virtual operating room of surgery for medical training of surgical nurses [10]. The proposed system integrates the use of a head-mount display, Oculus Rift. The experiment through the role-play with the system shows that the sense of presence and immersion has been enhanced in the majority of the participants. It is a good alternative for medical training. It is also used for anatomy lectures with 3D anatomical models.

III. Annotation of Sharing for 3D Models

In this work, we propose an annotation sharing and a viewpoint sharing mechanisms for 3D models. The former one is a mechanism for displaying annotations in consideration of the camera position. It means that an annotation to be displayed with a 3D model is stored with the position information of the camera, which decides the visual appearance of the 3D model. The system selects annotations of the 3D model properly based on the camera viewpoint changed by operations (i.e., rotation and zoom). An annotation does not use a link to express its context because the implicit line from the camera position to the 3D model going through the annotation takes the same role of the link. Thus, we realize the way of seamless display for 3D models with annotations.

The second one is a viewpoint sharing mechanism based on the Pub/Sub model. The mechanism provides viewpoint synchronization among multiple users who look at the same 3D model without mutual exclusion. Thus, it is capable of offering a scalable service.

A. Operations on 3D Models

Shared 3D models can be operated by users. This system suppose rotation and zoom as operations on 3D models. Rotation is defined as an operation of changing the angle θ without changing the distance l from the origin (see Figure 1). It is represented by $rotate(\theta, d)$, where d is the value of the angle to be changed from the current position. Rotation is primarily an operation for vertical and horizontal movement of the camera angle to change the face of the 3D model.

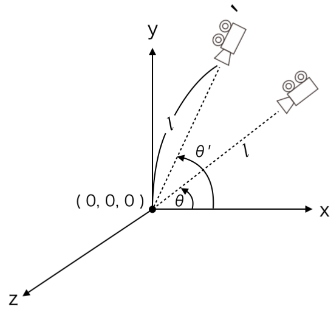

Figure 1: Rotation of a 3D model

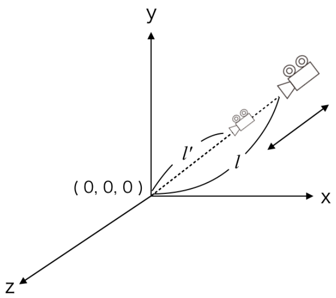

Figure 2: Zoom of a 3D model

Contrary to the rotation, zoom is defined as an operation for changing the distance l from the origin (see Figure 2). It is represented by $zoom(l, d)$. Zoom is to change the depth of the camera viewpoint to the model.

B. Annotation with Position Information

As mentioned above, the appearance of 3D models is different depending on the camera angle. Existing approaches introduced in Section II associated annotations with its context and semantics by using links explicitly. However, it makes the display complicated when there are lots of annotations on a 3D model in those methods.

We attempt to solve the problem using annotation with the camera position. Our approach selects annotations to be displayed properly according to the current camera angle. It also realizes implicit links between annotations and the 3D model to express their context.

1) Definition of annotations: When we consider the implementation, it is necessary to define the notion of the annotation with position information. First of all, we describe the definition of a set of annotations as follows.

 ISBN: 978-1-61208-538-8

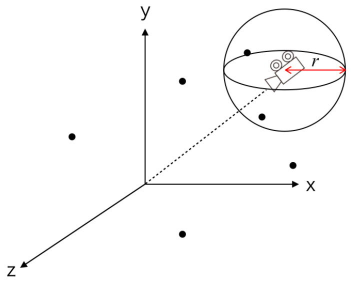

Figure 3: Selection of annotations

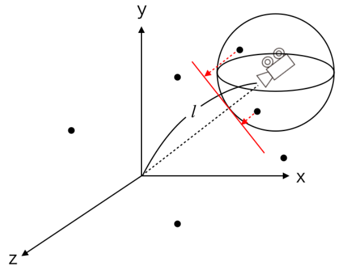

Figure 4: Converting 3D position information into 2D one by the Map function.

$$X = \{C_1, C_2, C_3, C_4, ..., C_n\} \tag{1}$$

In Equation (1), annotations are represented by the elements of X. Each annotation C_i is an ordered pair which consists of two elements $C_i \stackrel{\text{def}}{=} (s, c_i)$; s is an annotation content (e.g., a character string) and a three-dimensional coordinates of the position $c_i = (x_i, y_i, z_i)$.

2) Selection of annotations: We describe a method for selecting annotations on a 3D model. To display annotations, they are selected from the set of annotations X based on the current camera position a. Let $f_x(a, c_i)$, $f_y(a, c_i)$ and $f_z(a, c_i)$ be functions which return the distance between the camera position a and a position of a certain annotation c_i in x, y and z axis, respectively. So, we define $d(a, c_i)$ as the function to calculate the relative distance between a and c_i in a three-dimensional space, as follows.

$$d(a, c_i) = \sqrt{f_x(a, c_i)^2 + f_y(a, c_i)^2 + f_z(a, c_i)^2} \tag{2}$$

The set of annotations $D \subseteq X$ displayed on a screen is defined as follows.

$$D = \{C_d \in X | d(a, c_d) \leq r\} \tag{3}$$

Each annotation $C_d \in D$ is selected based on the sphere with the radius r from the camera position as the central point of it (see Fig. 3).

3) Display of annotations: Each annotation $C_d \in D$ should be converted its position c_d into two-dimensional one for showing on a flat display. So, we introduce a function $Map(c_d)$ for converting a position information. Figure 4 shows how it works. The advantage of this function is to preserve the relative position between annotations in a 2D plane. When a user rotates the 3D model (i.e., changes the camera position), positions of annotations in a display should be updated. The Map function updates those positions for following such operations as shown in Figure 5. As a result, annotations move to the same direction of the rotation.

4) Solving the scale problem: In the case that there exists a gap between the size of the 3D space and the display size as shown in Figure 6, we have to adjust the difference between them properly. The $Comp$ function

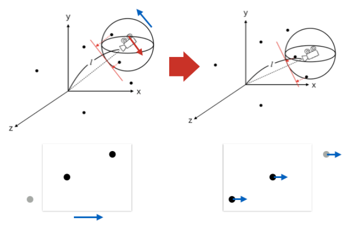

Figure 5: Updating positions of annotations in a display.

defined below adjusts the positions of annotations for the display to solve the scale gap (see Figure 7).

$$Comp(c_i) = (W + k_w x_i, H + k_h y_i) \tag{4}$$

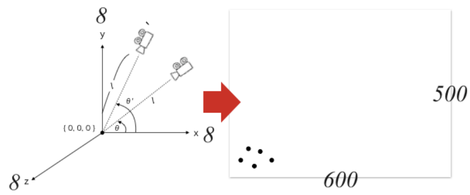

Figure 6: The scale problem caused by the gap between the size of the 3D space and the display.

W is the half of the width and H is the half of the height in the display. k_w and k_h are correction factors for x-axis and y-axis, respectively. They are calculated by the following equation:

$$k = \frac{L + \frac{L}{2}}{|l_{min}| + |l_{max}|} \tag{5}$$

Let L be the width or the height of the display area. l_{min} and l_{max} indicate the minimum and the maximum of

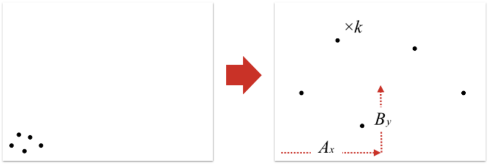

Figure 7: Position adjustment for annotations by the *Comp* function.

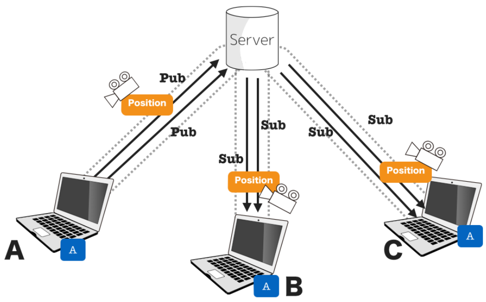

Figure 8: Sharing viewpoint based on the Pub/Sub model

the 3D space (i.e., $\forall x, y, z \in [l_{min}, l_{max}]$). To obtain k_h, you set the height of the display as L. While, you can set the width of the display as L to obtain k_w.

IV. Viewpoint Sharing for 3D Models

We propose a mechanism of viewpoint synchronization among multiple users for shared 3D models. We apply the Pub/Sub communication model, which is mostly used in distributed system community [11], [12], to the mechanism to solve the performance degradation.

Suppose that all users in Figure 8 have registered to the server to subscribe the same 3D model. In the viewpoint sharing mechanism, the current viewpoint information (i.e., camera position) of a user is published to the server when the user makes some operation to a shared 3D model (i.e., the user A in Figure 8). While, other users behave subscribers in the Pub/Sub model. It means that such users receive the camera viewpoint on the shared one.

Advantage: The advantage of this approach is that it provides viewpoint sharing for 3D models in a scalable manner. Thus, we expect that it is capable of offering real-time service to multiple users. Moreover, a camera viewpoint on a 3D model will be delivered to appropriate users based on the registration information which associates users (or clients) and 3D models.

Applications: Our proposed system including viewpoint and annotation sharing mechanisms can be useful, for instance, for teleconference systems in medical practice. 3D models of internal organs which are obtained by fMRI are used for preparing or practice for surgical operations. In this case, annotation and viewpoint sharing can be building blocks of teleconference systems for a meeting using 3D models among multiple sites (hospitals). Practically speaking, we can build a teleconference system using 3D models with our mechanism of annotation and viewpoint sharing together with some communication service. It is also possible to implement a system for distance learning with molecular structure models using the proposed system. 3D models can be helpful to understand chemical structures, molecular dynamics, universal gravitation among planets.

V. Details of Implementation

In this section, we explain the implementation details of our prototype for sharing annotation and viewpoint on 3D models.

The architecture of our prototype is shown in Figure 9. The prototype consists of the server and clients. Clients have been implemented on the browser Google Chrome version 54. We describe the details of them below.

1) Implementation of the server: The server is implemented using the Web application framework Express of Node.js and Socket.io that is a module related to WebSocket. Socket.io provides the push-style communication interface between the server and clients. If a user changes the camera position by rotating the 3D model, this information is sent to the server immediately. It is like *publish* in a Pub/Sub system. The server manages the subscription information and 3D models in Redis which is an in-memory Key-Value-Store (KVS) supporting pub/sub-style messaging.

2) implementation of client side: Clients register a certain 3D model and *subscribe* the change of the viewpoint from the server. In this architecture, a client that executes a certain operation (e.g., rotation or zoom) becomes a *publisher* and any other clients are *subscribers*. We adopt WebGL for displaying 3D models on a browser. In the implementation of clients, we mainly use Three.js which is a wrapper library of WebGL. A Client implemented on a browser is provided as an extension of the browser (i.e., add-on). This is because, we suppose that the proposed system is used with some messaging service such as Google Hangouts, Skype and so on.

3) Advantage of the prototype: The advantage of this messaging style is that a publisher does not care about the destination. It sends the viewpoint information to the server as a message after executing an operation, and the server delivers it to subscribers properly. Moreover, Redis takes place the pattern matching between published messages and subscribers. Thus, our prototype can execute the view synchronization with low overhead.

VI. Evaluation

We have measured our proposed mechanism of the viewpoint sharing mechanism from two points of view; the resource consumption in the server and the response time in clients. The former one would be useful for system administrators, or service providers that provide some

 ISBN: 978-1-61208-538-8

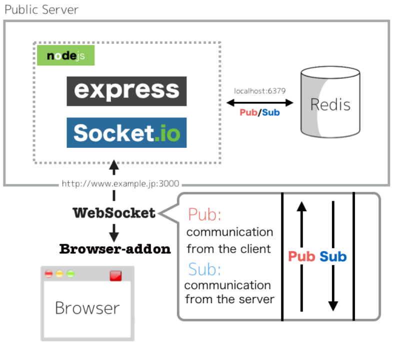

Figure 9: Architecture of the system

service using our proposed mechanism and the latter one might be interesting in terms of user experience.

A. Environment and Experimental Setup

Our experiments have been done in a LAN including a single server and 20 clients where they are interconnected with a wired 1000Base-T Ethernet connection. The average RTT (round trip time) is 0.71 msec. in the LAN. Clients run on Mac OSX version 10.9.5.

The number of clients is set from 2 to 20 in the experiments. We used the regression testing tool Selenium Web Driver running on web browsers to realize a heavy load to the server during the experiments.

Both the average of CPU load and Network load were measured using the System Admin Reporter (sar). We have measured the number of received packets and number of sent packets and the CPU utilization of the server process for 200 sec. for each configuration. We have also measured the response time between the server and clients as observing the time difference between a pair of sent and received packets in clients.

B. On Resource Consumption

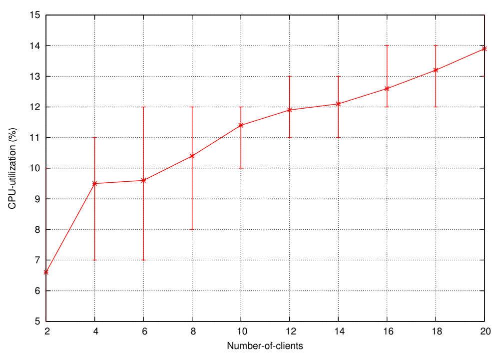

Figure 10: The CPU load of the server

1) CPU Load: Figure 10 shows the result of the number of clients v.s. the average value of CPU load (%) at the server. The average CPU load of the server process is shown in the y-axis, and the number of clients is shown in the x-axis.

The average CPU load of the server is 13.9% with 20 clients. It is increased by about 2.1 times from the one with 2 clients. If we suppose an increase of the CPU load is same as our experimental result, the average CPU load is expected to be about 43% with 100 clients.

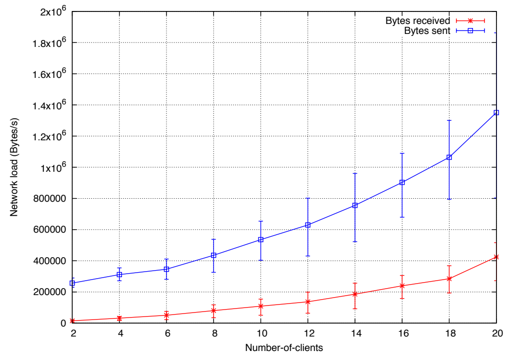

Figure 11: Network load on the server

2) Network Load: Figure 11 shows the result of the number of clients v.s. the average network load (Bytes/s) at the server. The number of sent and receive packets is shown in the y-axis, the number of clients is shown in the x-axis.

Packets received in the server increases 150 Kbytes/s to 323 Kbytes/s as the number of clients increases from 2 to 20. On the other hand, packets sent from the server becomes 1,067 Kbytes/s from 256 Kbytes/s by the number of clients increases. As a result, the architecture of the experiment can be practical with 1,000 clients because the number of packets is expected to be 40 Mbytes/s.

C. On Response Time

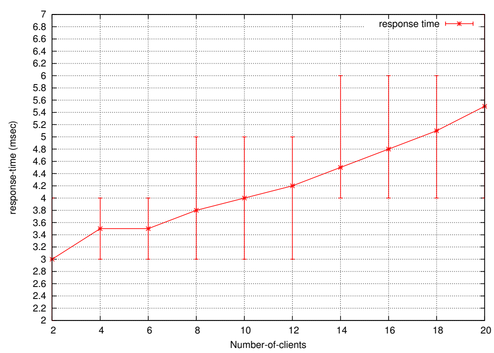

Figure 12: Response time on the number of clients

Figure 12 shows the time for processing a request of updating a viewpoint at the server. We call it the response time.

According to the result, the response time is expected to be about 100 msec if we suppose 700 clients. 100 msec. is considered to be the limit of response time for having users feel that the system is reacting instantaneously [13]. Thus, our prototype is capable of having the real-time property with about 700 users.

VII. Conclusion

We showed our prototype for sharing annotation and the viewpoint synchronization mechanism based on the Pub/Sub model for 3D models in this paper. It realizes the scalable collaboration using 3D models for multiple users.

We also measured resource consumption of the server and response time in clients on the viewpoint synchronization mechanism. As a result, the system showed the capability of providing the real-time service to at least 100 users. It means that the prototype can provide the high-level user experience with low overhead.

The proposed system, for example, can be used for teleconferencing in the medical field. It is useful for an online meeting using a 3D model of an internal organ, which is the target for a surgical operation. It can also be used for e-Learning in physics and chemistry. The simulation of planetary motion and chemical synthesis using the prototype provides an additional learning experience for students.

In our project, we are developing the collaborative editing system for 3D models using a conflict-free data type, called ChainVoxel [14]. The next goal of the work is to combine the annotation and the view synchronization mechanism with the collaborative editing system and develop the integrated framework of 3D models for e-Learning, teleconferencing, digital archiving, city design, and training in the medical field.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16K00449.

References

[1] J. Jang-Jaccard, S. Nepal, B. Celler, and B. Yan, "WebRTC-based video conferencing service for telehealth," *Computing*, vol. 98, no. 1, pp. 169–193, 2016.

[2] T. Jung, E. Y.-L. Do, and M. D. Gross, "Immersive redlining and annotation of 3d design models on the web," in *the 8th International Conference on Computer Aided Architectural Design Futures*, pp. 81–98, 1999.

[3] T. Jung, M. D. Gross, and E. Y.-L. Do, "Annotating and sketching on 3d web models," in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, ser. IUI '02. New York, NY, USA: ACM, pp. 95–102, 2002.

[4] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. R. Swick, "Annotea: An open rdf infrastructure for shared web annotations," in *Proc. of the 10th International World Wide Web Conference*, pp. 623–632, 2001.

[5] R. Swick, E. Prud'Hommeaux, M.-R. Koivunen, and J. Kahan, "Annotea protocols," W3C, 2002, http://www.w3.org/2002/12/AnnoteaProtocol-20021219#AnnotProtocol retrieved: March 2017.

[6] R. Kadobayashi, J. Lombardi, M. P. McCahill, H. Stearns, K. Tanaka, and A. Kay, "3d model annotation from multiple viewpoints for croquet," in *Proc. of The Fourth International Conference on Creating*, ser. Connecting and Collaborating through Computing, pp. 10–15, 2006.

[7] J. Hunter and C. hao Yu, "Supporting multiple perspectives on 3d museum artefacts through interoperable annotations," in *Cultural Computing*, R. Nakatsu, N. Tosa, K. W. Wong, and P. Codognet, Eds. Springer Berlin Heidelberg, pp. 149–159, 2010.

[8] J. Hunter, M. Henderson, and I. Khan, "Collaborative annotation of 3d crystallographic models," *Journal of Chemical Information and Modeling*, vol. 47, no. 6, pp. 2475–2484, 2007.

[9] P. T. Goeser, W. M. Johnson, F. G. Hamza-Lup, and D. Schaefer, "View: A virtual interactive web-based learning environment for engineering," *Advances in Engineering Education*, vol. 2, no. 3, pp. 1–24, 2011.

[10] N. F. Kleven, E. Prasolova-Førland, M. Fominykh, A. Hansen, G. Rasmussen, L. M. Sagberg, and F. Lindseth, "Training nurses and educating the public using a virtual operating room with oculus rift," in *Proc. of the 2014 International Conference on Virtual Systems and Multimedia (VSMM)*, pp. 206–213, 2014.

[11] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, Jun. 2003. [Online]. Available: http://doi.acm.org/10.1145/857076.857078

[12] Y. Liu and B. Plale, "Survey of publish subscribe event systems," Indiana University, Tech. Rep. TR574, May 2003.

[13] J. Nielsen, *Usability Engineering*, 1st ed. Morgan Kaufmann, 1993.

[14] K. Imae and N. Hayashibara, "ChainVoxel: A data structure for scalable distributed collaborative editing for 3d models," in *Proc. of The 14th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC-2016)*, 2016.