VizGrid: Collaborative Visualization Grid Environment for Natural Interaction between Remote Researchers

● Ryuichi Matsukura ● Koji Koyamada ● Yasuo Tan
● Yukihiro Karube ● Mitsuhiro Moriya

(Manuscript received September 8, 2004)

Recently, as the borders between research fields have become fuzzy, there have been increasing opportunities for researchers in different fields to collaborate with each other. VizGrid is a joint research project between universities, government institutions, Fujitsu, and other industry members that was partially founded by the Ministry of Education, Culture, Sports, Science and Technology. This project aims to develop a new collaboration space in which three-dimensional (3D) images can be manipulated in a highly realistic virtual environment. This paper describes Volume Communication, which is an infrastructure we developed for the VizGrid project that can generate, communicate, and display functions of 3D image data. We also describe the technical problems in transmitting high-resolution 3D images based on the evaluation of the prototype system we developed. The collaboration space proposed in this paper is used to communicate common objects that configure a virtual environment and images of humans as 3D image data. The receiving side visualizes this data so that important functions, for example, eye-to-eye contact, can be realized to simulate a real face-to-face environment.

1. Introduction

Recently, as the borders between research fields have become fuzzy, there have been increasing opportunities for researchers in different fields to collaborate with each other. Moreover, the most advanced research often requires special facilities and equipment. In those cases, researchers have to collaborate with other researchers at remote sites to use the facilities. In a collaborative research that extends over several fields, the same word can have subtle or even large differences in meaning and can be perceived differently by researchers in different fields when telephones and videoconference systems are being used. This often results in misunderstanding or confusion among researchers. In a collaborative research environment, therefore, mutual understanding is a key factor. Especially, for collaborative research carried out by researchers in different locations, a collaboration environment that employs images capable of conveying realistic sensations will be required.

Grid computing[1] has been rapidly spreading as a technical foundation for this field. Grid middleware such as the Globus Toolkit[2] and UNICORE[3] connects a number of distributed resources for computing and provides functions for searching and monitoring the distributed resources as well as a security infrastructure that includes authentication. The middleware also provides the means for accessing the distributed data and executing applications at the remote sites.

VizGrid[4)-6)] is a joint research project between universities, government institutions, and Fujitsu that was partially founded by the Ministry of Education, Culture, Sports, Science and Technology (MEXT). This project is responsible for creating a new collaboration space in which 3D images can be shared for the remote collaborative research described above. We are engaged in the development of elemental techniques for transmitting 3D images and also in the realization of a remote collaborative research environment that employs computer simulations for medical services and experiments in nuclear fusion. The VizGrid applications are designed to use Grid technology to process visualization for the enormous amounts of data generated from computer simulations for medical services and experiments in nuclear fusion. For the collaborative research environment, however, not only a reduction of the total computing time but also natural and smooth interactions must be considered.

This paper describes Volume Communication, which is a software infrastructure for 3D communications we developed for the VizGrid project, and explains the actualization procedures. Based on the evaluations of a prototype system we developed, this paper also describes the technical problems in transmitting high-resolution 3D images as well as the functions required for a Grid environment. This paper then describes a collaboration environment that employs a Grid environment.

2. Communications using 3D images

For the purpose of intelligence sharing, we have so far made our investigations in the study of how to represent 3D objects on 2D planes simple and clear. In construction projects, for example, drawings used by architects are designed so that the people handling the construction work can easily understand them. Also, tomographic images such as CT/MRI images are designed to aid doctors in making diagnoses. However, these 2D images are created with a focus on specific parts that are important for the specialists in the relevant field. There is no guarantee that specialists in other fields will interpret the images in the same way. In the remote collaboration environment that VizGrid is intended for, specialists can hold discussions while sharing 3D images (**Figure 1**). By employing 3D images, specialists can make the most of their intuitive faculties. Moreover, a virtual natural environment can be created in which they feel they are having face-to-face interactions.

Meanwhile, development of the hardware required for displaying 3D information has been making progress. In Japan, the Consortium of 3D Image Business Promotion[7] and the 3D Consortium[8] were established in 2003 to familiarize people with 3D display technology. Several technological approaches have been proposed, and products based on these approaches are already available on the market. We will not only develop applications to use these technologies but also offer a software infrastructure called Volume Communication to develop applications for 3D image transmission.

There are several ways of representing 3D data, and VizGrid does it by using volume data. When transferring a 3D visualization image to a remote site, applications generate an isosurface (a 3D surface consisting of points, each of which is described by a single scalar value) from volumetric data. Most applications then transfer the isosurface as polygonal data. However, when transferring polygonal data, the application must

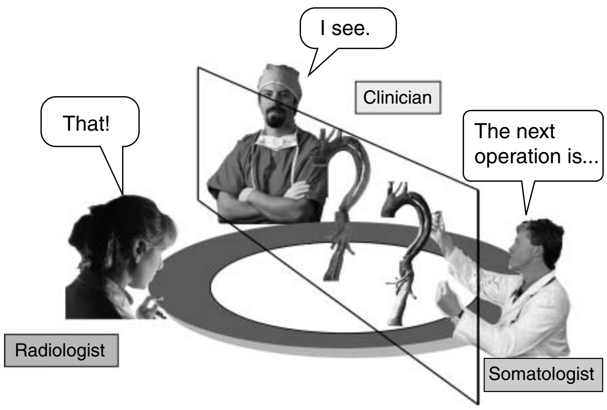


Figure 1
3D collaboration environment envisioned for VizGrid.

retransfer an entire image whenever the parameters for visualization are changed. Moreover, when people in multiple sites look at the same 3D images, they cannot see them from their own individual viewpoints, because the sending site only generates images from a single viewpoint.

For remote collaboration, it is important for participants to see images from their own viewpoint. Therefore, we choose volume data as the data format for transmission. In VizGrid, to enable efficient visualization processing and easy estimation of the resources required for transmission, we employ a volume that contains voxels (volume pixels) in a cubic grid as the basic data unit. The voxels of volume data have basic information such as color, brightness, and a rate vector.

Volume Communication is a concept for not only constructing an environment for communicating 3D data, but also for realizing a remote cooperative research environment in a virtual space created using 3D images. The core techniques of Volume Communication include generation, compression, transmission, decompression, archiving, searching, and display of volume data and other supportive techniques (**Figure 2**). The targets of the volume data generation technique are the simulation results, CT/MRI measurement results, and data reconstructed from several camera images. The volume data compression and decompression techniques are based on the characteristics of 3D data and have a real-time performance suitable for interactive communications. The transmission techniques consist of 1) protocols for transmitting volume data as streaming data and correcting any errors it contains and 2) a signaling control protocol that enables several sites to start communication with each other. The archiving and search techniques generate indexes according to the characteristics of the 3D data in the archiving phase and retrieve 3D data from archives using these indexes. The display technique is a high-speed visualization technique that converts volume data into images in real time. This display technique is used because conversion to 2D images is required even for stereoscopic display devices.

3. Software infrastructure for Volume Communication

This section describes in detail the flow of operations from the volume data generation to the transmission and display required to realize a 3D image collaboration environment among the Volume Communication platforms.

3.1 Generation

All of the data handled by the framework of Volume Communication must be converted to a volume format. The volume data generation func-

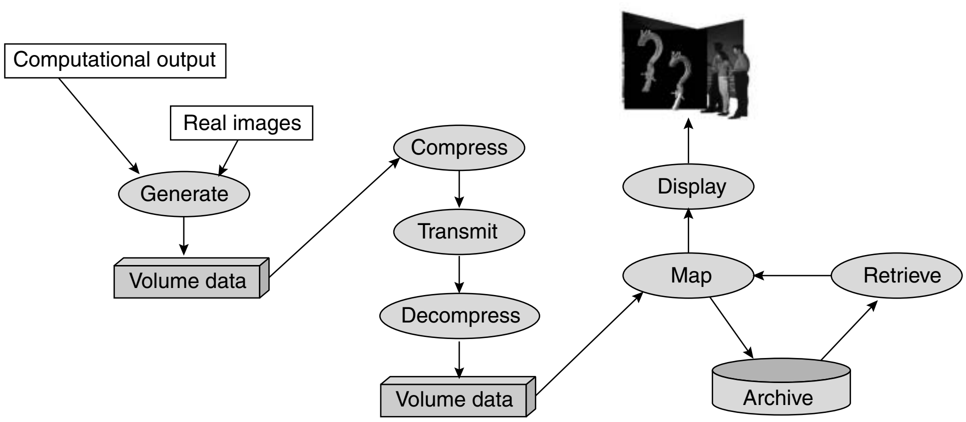


Figure 2
Volume Communication suite.

tions handle this process. There are two types of 3D image data. One is based on data already generated in computers; the other is based on data obtained by measuring objects in the real world. An example of data generated in computers is artificial data such as simulation results; 3D images created using CAD also fall into this category. Because data within a computer can easily be converted to a volume format, this type of data is probably the most used in collaboration spaces. An example of data obtained by measuring is the data obtained from the CT/MRI equipment used in medical facilities. This equipment takes continuous tomograms of slices several millimeters thick to obtain an image. Further analysis and simulation are then performed after extracting the required parts from the series of images. An example application of CT/MRI is the extraction of the shape of an aorta reconstructed from CT images and simulations of blood flow.[9]

In the prototype system developed in early 2004, five video cameras are used to take images of the same object from different angles. A 3D shape is then reconstructed from the images, converted to volume data, and transmitted. To reconstruct the 3D shape, a visual cone intersection method is used to render the 3D shape as a collection of voxels. Next, voxel coloring is performed to color each voxel without generating a triangle mesh. **Figure 3** shows the flow of processing.

A visual cone intersection method capable of being performed in real-time is effectively used for the 3D reconstruction method using video cameras at multiple viewpoints. This method enables the structure to be output as voxels and directly colored based on the image data without converting the voxels to a triangular mesh. For the visual volume crossing method, parallelization is performed using PC clusters. VizGrid employs high-speed algorithms developed by Professor Matsuyama of Kyoto University.[10] For the generated voxels, the color data of the corresponding sections photographed by each camera is extracted and averaged to determine the colors of the voxels.[11]

3.2 Transmission

New technology is required to transmit the 3D motion image data. It is required not only because the amount of data has increased (because the objects to be transmitted are now 3D instead of 2D), but also because the volume of computation needed to calculate the 3D positional

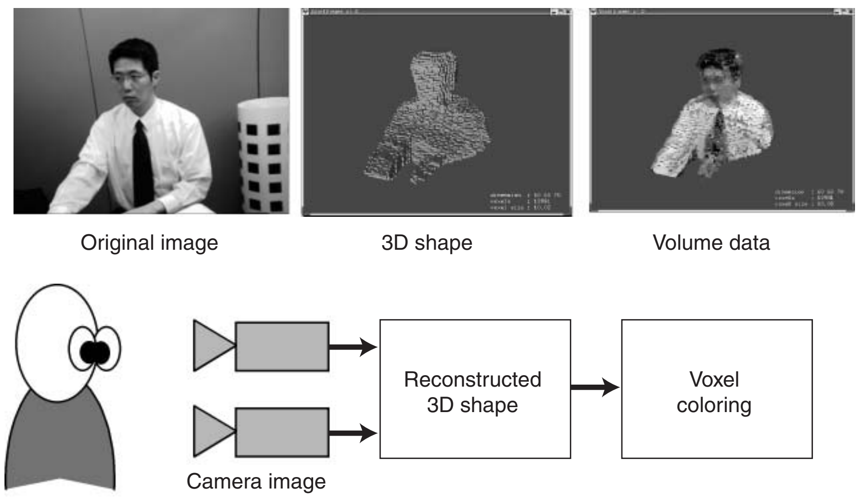


Figure 3
Volume data generation from multiple calibrated cameras.

relationships has increased. Even 2D transmission requires technology to synchronize the different types of media and stack and display the images. For 3D transmission, however, it is necessary to code the exact positional relationships. More precisely, synchronization among the media is also required. For VizGrid, we considered using a unified protocol to transmit the streaming data transmitted by Volume Communication. For the streaming data, we have taken into consideration the need to unify the sequences of coordinates of the pointing device and general-purpose serial data in addition to the volume data. By unifying the handling of the streaming data, the communication synchronization, positional relationships, and in particular the coordinate points can be matched relatively easily.

For the transmission method, there is no standard protocol at present that can handle streaming transmission of the 3D data. We have investigated a method for transmitting the volume data in real time. The difficulty in transmitting volume data lies in the fact that it is large compared with 2D images. For example, to transmit volume data having a resolution of 512 × 512 × 512 voxels with a frame speed of 15 fps (frames per second), a bandwidth of 60 Gbps is required. To enable this transmission, we have considered various techniques such as reducing the required bandwidth using real-time compression and decompression and dividing the volume data into several smaller volumes and transmitting them using separate networks. For the compression method, we have investigated various methods, for example, vector quantization. Real testing, however, shows that compression using a Pentium 4 PC operating at 3 GHz requires a processing time on the order of several minutes to several hours. Therefore, at present, processing compression of 3D data in real time requires hardware support.[12] For VizGrid, we plan to use software to perform the compression, giving priority to high-speed processing over an increase in the compression rate. We expect to employ a transmission method that divides the volume data into packets.

When transmitting divided volume data or receiving data from multiple sites, delays on the network will lead to a loss of synchronization. Normally, the terminals used for videoconferences and other such meetings have internal functions that synchronize the audio and video. However, VizGrid uses PC clusters to enable distributed processing of applications. As a result, a complex architecture is required to achieve synchronization at the receiving side. We therefore developed a system for synchronizing multiple streaming data on the network nodes.[13]

3.3 Visualization

We are developing visualization techniques that focus on real-time visualization of 3D motion pictures and a collaboration environment based on volume rendering that makes the users feel as if they are collaborating in an actual 3D environment.

The most important factor regarding volume rendering is the real-time capability. The processing time required by algorithms for general-purpose volume rendering depends on the resolution involved. Therefore, the balance between the rendering speed and resolution required by the collaboration environment must be considered.

We developed two new technologies. The first is for high-speed rendering using the solid texturing facility of a PC's graphics hardware. The problem here is that the procedure for volume rendering becomes inefficient when there is insufficient real memory in the graphics hardware. To solve this problem, the first new technology divides the massive amounts of volume data into smaller amounts, which can then be loaded into the memory of the solid-texturing hardware and processed one at a time. Then, the partially visualized images are overlapped in the proper order to produce the complete visualized image.

The second new technology uses a new sampling method for volume rendering. In conventional volume rendering using solid texturing, the volume data is sliced into equidistant surfaces, the surfaces are visualized into images, and then the images are overlapped. However, when a detailed representation of an internal structure is visualized using this method, the edges of the generated images are more deteriorated than their centers. Our system avoids this deterioration by slicing the volume data to be visualized so it has a spherical surface.

4. Prototype

We developed the prototype system as a primitive implementation for the Volume Communication platform in early 2004. This section describes the prototype system and the issues involved in producing a practical system.

4.1 Specifications

Figure 4 shows the prototype system configuration. For the prototype system, volume data was created from images taken by five different cameras. A 3D image was transmitted at the rate of two frames per second. An image of two different viewpoints visualized by two PCs was then displayed from the same image data. The resolution of the 3D image was 96 × 96 × 96 voxels. As shown in **Figure 5**, five cameras were secured to a pipe frame measuring 3 × 2.2 × 2 m. To reconstruct a 3D shape from camera images, objects are extracted from the background using a background subtraction method. The surrounding area was curtained off to create a plain background so the object could be easily extracted. The 3D structure of the object in an area measuring 1 m high × 1 m across near the center of the frame was reconstructed and converted to volume data. The size of the basic voxel was 1 × 1 × 1 cm.

To create the volume data from the camera video, six PC clusters (CPU: Intel Xeon 3.06 GHz) were used. The cameras were connected to the PCs using IEEE1394. The PC clusters were interconnected using a gigabit Ethernet. One of the six PC clusters was used for control, and the other five were used for parallel processing to create the volume data.

For the transmission section, the volume data was transmitted using RTP (Real-time Transport Protocol). SIP (Session Initiation Protocol) was used for connection control. In the prototype system, the volume data was divided into 1500-byte packets with approximately 3500 packets comprising one frame. Because the receiving side was connected using a gigabit Ethernet, packet loss was extremely low. Therefore, we did not

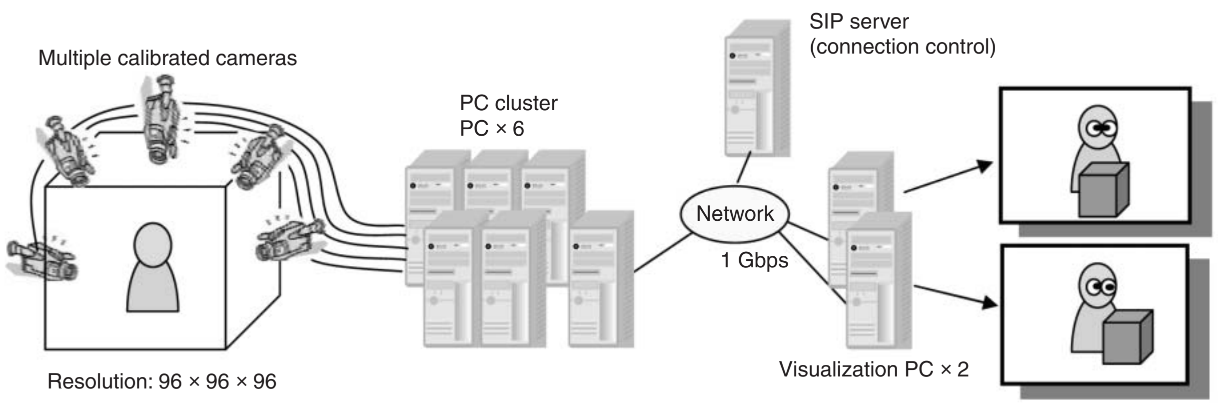
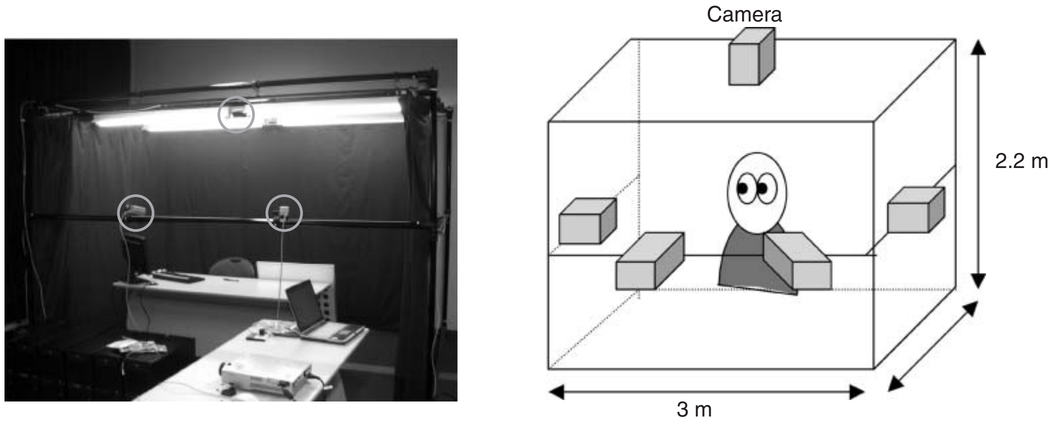


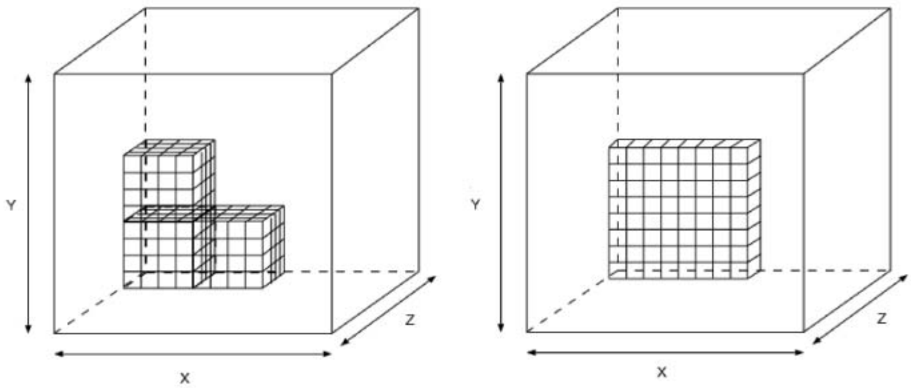
Figure 4
Prototype system configuration.



(a) Studio for capturing 3D motion pictures (Pipe frame securing the cameras)

(b) Camera locations on the pipe frame

Figure 5
Studio and camera locations.



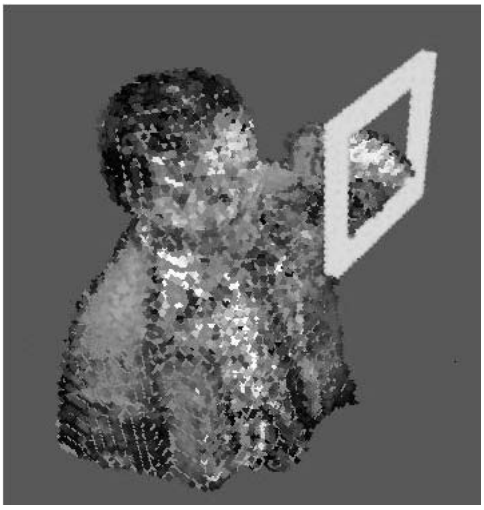
(a) Cube division: Divided into cube/ rectangular parallelepiped blocks

(b) Surface division: Sliced at any surface

Figure 6
Volume division methods.

incorporate error correction into the prototype system. If, however, packet loss occurred within a frame, the frame was abandoned. In the test environment we constructed, the percentage of frame abandonment was about 1%. If the packet loss rate becomes too high on a specific network, error detection and correction functions can be added when the data to be transmitted is organized into packets. These functions would restrict the range of data that is affected by a packet loss and perform interleaving for the handling of burst errors. The affected range of data can be restricted by transmitting the volume data in small packets. We are investigating two methods of dividing the volume data into packets: dividing using slices and dividing using cuboids (**Figure 6**).

To make the images, we employed a volume rendering method developed in 2003. We selected this particular type of volume rendering because it can be used as a platform to perform a wide range of operations. Simple CAD data is overlaid to form the images of the prototype system. **Figure 7** shows an example of a person's arm extended into the square frame. To synthesize the images, we developed functions that



Simple CAD images layered onto a person

Figure 7
Images visualized by the prototype.

overlay multiple 3D objects within a space of 96 × 96 × 96 voxels. For example, if we photograph the upper half of a body, the resolution is 60 × 60 × 70 voxels, which enables the overlay of objects that have a resolution of about 20 × 20 × 20 voxels. After synthesizing the image using 3D data, the image is converted to a 2D image by volume rendering. In this way, we can represent interactions between a person and an object.

Under the current installation, the size of the 3D space is fixed, and further investigation will be required to increase the overall resolution through the synthesis of multiple objects. This is because increasing the resolution presents certain problems. For example, communication over a single network will become more difficult due to the increased amount of transmission data. Also, the amount of rendering required will increase, causing real-time processing to become more problematic.

4.2 Improving the prototype system

The current prototype is not yet a practical collaboration system. The following section describes how it can be improved, especially the resolution of volume data that can be transmitted and the connection control of multiple sites.

4.2.1 Increasing resolution of transmitted volume data

The transmission data is not compressed in the prototype because we did not take the current resolution of the transmission data, the PC performance, and the network bandwidth into consideration. The transmission rate of the prototype is about 60 Mbps. When the resolution is increased to produce high-quality visualized images, it becomes difficult to transmit them at high speed. Under the current conditions, software compression cannot be used for real-time communication unless it is incorporated into the hardware and its speed increased. This means that a data compression method that exploits the characteristics of the data to be compressed must be investigated. For example, for data reconstructed from camera images, only the surface data of the object needs to be reproduced, because there is no data for the interior. For data reconstructed from images taken by cameras from multiple viewpoints, therefore, the data size can be reduced if we just transmit the parts that contain numerical values. In the human image shown in Figure 7, approximately 10 000 voxels have numerical values, which is only about 1% of the overall image. When reconstructing 3D shapes, the prototype receives images whose resolution is 320 × 240 pixels, which is one-fourth the original resolution of the camera images. Therefore, it is possible to increase the space resolution. We are investigating an algorithm to reduce the processing time for 3D reconstruction. In addition, compared to natural images, data used for visualizing simulation data has fewer restrictions and correlations; as a result, we can increase the degree of data compression for this type of data in the current collaboration environment.

The current resolution is restricted by the PCs' computational capabilities and the network bandwidth. The prototype system can handle a resolution of 96 × 96 × 96 voxels, and the maximum frame rate is 2 fps. Because the data is not compressed, the limit on performance becomes apparent when large amounts of communication data are transmitted. In addition, to use the system as a collaboration system, advanced functions that not only render images of people reconstructed from camera images but also overlay them with 3D images are required. The current collaboration space of VizGrid requires an environment that combines images from three or more viewpoints.

4.2.2 Multisite connection

For VizGrid to realize a collaboration environment that can closely simulate an actual face-to-face meeting, we have proposed a collaboration space that positions the images of people as if they were surrounding the object under discussion (Figure 7). We feel that this space can be achieved by combining multiple 3D image transmissions using Volume Communication as described above. **Figure 8** shows the overall configuration of our proposal. The 3D images photographed and reproduced at the individual sites are reproduced and displayed as images viewed from each site while maintaining the positional relationships.

For the objects used for the collaboration work, overlay processing is performed at the sending side. However, using one 3D space to represent collaboration among multiple individual sites requires that a single site be in charge of managing the volume data present in the collaboration space. For the framework of Volume Communication, the volume data sent from the sites is combined to represent the collaboration space. Precise coding of the positional relationships of the individual volume data items and reconstruction of the im-

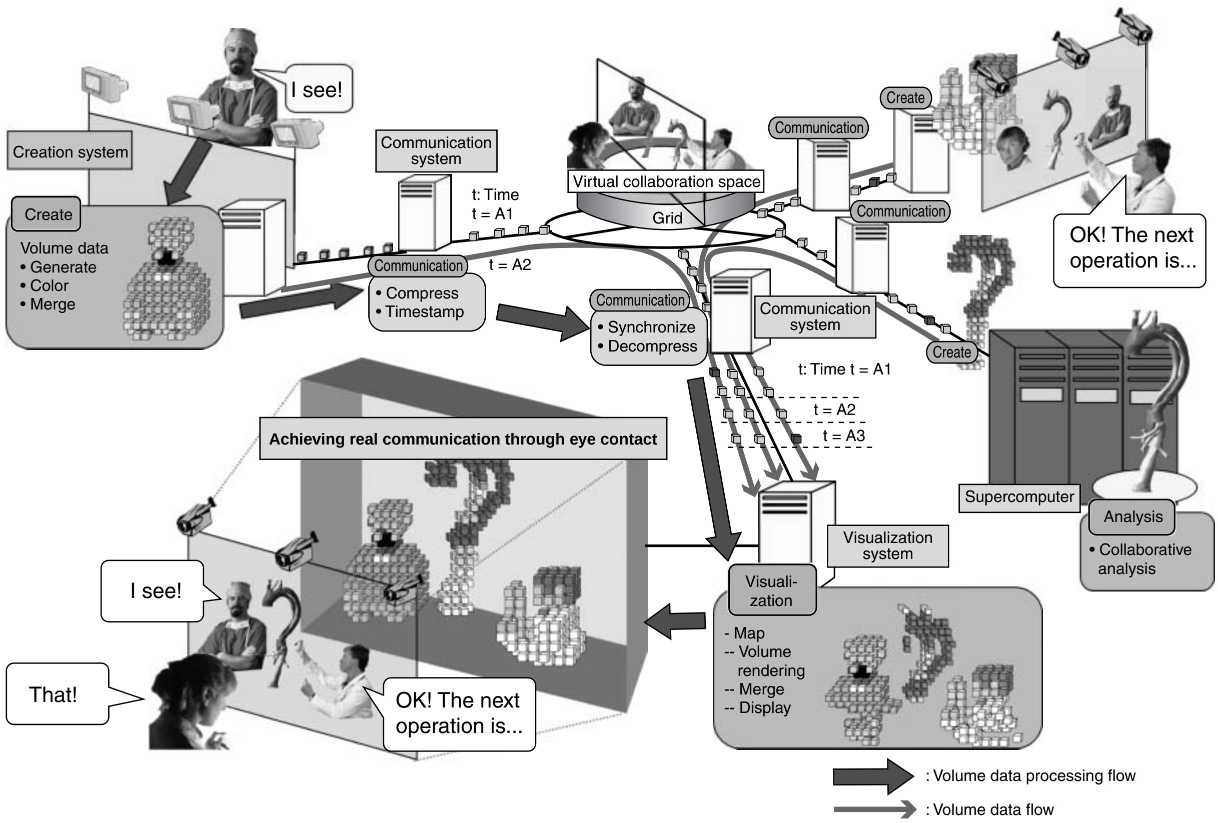


Figure 8
3D collaboration space.

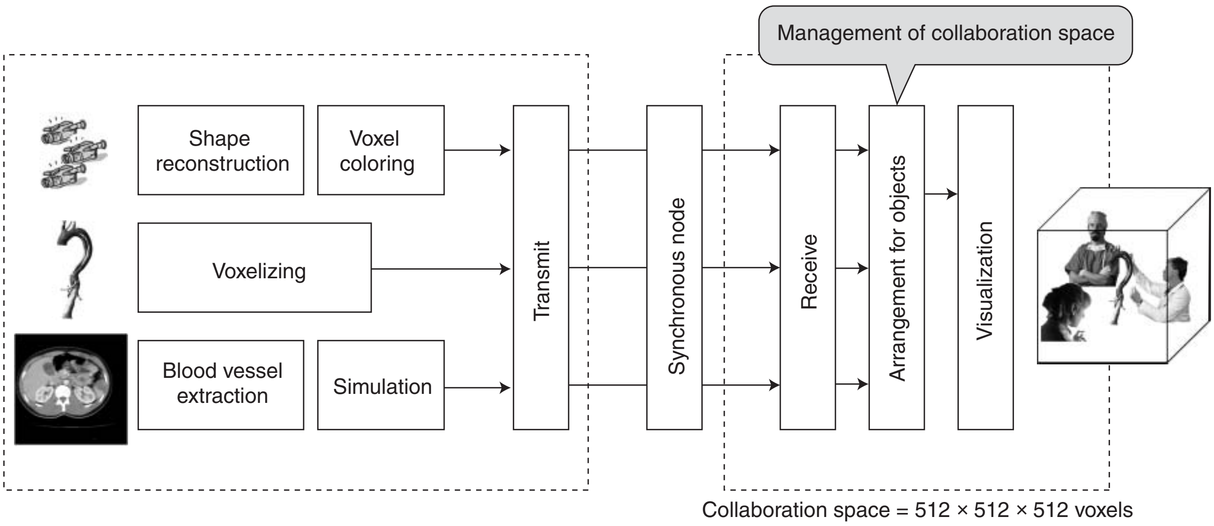


Figure 9
System configuration achieving 3D collaboration including information sharing.

ages that reflect these positional relationships enable precise representation of the participants' actions. That is, eye contact between the participants and the items being pointed to by the participants' hands and fingers can be shown accurately, resulting in a greater feeling of presence at the scene of interaction.

To represent interactions precisely, the events that are generated by the interacting objects must be synchronized. For example, to represent the pointing action of a person in conjunction with the movement of an object, the event and pointing action must be synchronized. However, synchronization presents its own problems. For example, if the image of the object is delayed or its display is speeded up, synchronization with the pointing action will be lost and a different event will be displayed. Therefore, we have developed a multiple streaming synchronization system that ensures continuous communication. In our system, the object images and people images are handled by different volume data streaming (**Figure 9**). As a result, the synchronization system enables the data to be synchronized on the network before it arrives at the receiving system. Nevertheless, the time required to make the images depends on the complexity of the object. Because there is no way to completely eliminate the causes of lost synchronization, it is imperative that when the images of objects are made, the objects that require synchronization are allocated a large amount of CPU resources. Moreover, if the intercommunication takes too much time, delays in interactions between interacting objects will occur and the collaboration will be unnatural. To handle the send resources separately, it is essential to use QoS (Quality of Service) and other services that prioritize the processing of send resources that require synchronization.

5. Conclusion

We have described a 3D collaboration environment for promoting collaborative research among different scientific fields and a method for putting the collaboration environment into practical use. For the Volume Communication platform offered by VizGrid, we have proposed using a volume format to represent 3D data and also proposed methods for generating, transmitting, imaging, archiving, and retrieving volume data.

In early 2004, we developed a prototype

system capable of transmitting 3D images reconstructed from five camera images as 3D animations at 2 fps with a resolution of 96 × 96 × 96 voxels. Moreover, we developed a platform for realizing the functions of Volume Communication, which is a software infrastructure for 3D communications we propose, and investigating the limits and possibilities of Volume Communication. To give an indication of the required transmission speed for 96 × 96 × 96-voxel 3D data, when the frame rate is 10 fps and there are 3.5 Mbytes per frame, the minimum required transmission speed is 280 Mbps. Therefore, when multiple streams are transmitted simultaneously, there will be insufficient communication bandwidth and a data compression methodology that overcomes this problem is urgently required.

To fully realize a collaboration environment, it is essential that the information be managed by a single collaboration environment. By using a Grid to realize the collaboration environment, the local environment does not need an imaging section. For the access lines, only the 2D images to be viewed at the site need to be transmitted. Thus, the part that has the bandwidth needed for computations can be incorporated as part of the Grid platform.

We plan to develop a prototype system that overlays medical images and camera images for applications in the medical field by 2005. We will also develop a system that can transfer 3D animations at 10 fps with a resolution of 256 × 256 × 256 voxels and at several fps and 512 × 512 × 512 voxels before the completion of this project in 2007.

Acknowledgement

We would like to thank Prof Matsuzawa, a project leader, and all the members of the VizGrid project for their useful comments. The VizGrid project is partly supported by MEXT in Japan.

References

1) I. Foster and C. Kesselman: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Pub, 1998.
2) Globus. *http://www.globus.org/*
3) UNICORE. *http://www.unicore.org/*
4) VizGrid. *http://www.vizgrid.org/en/index.htm*
5) M. Okuda, H. Morishige, Y. Karube, M. Yamada, H. Shitara, and T. Matsuzawa: VizGrid Development of Teleimmersive Collaboration Environment. Proc. 3rd PDCAT, Kanazawa, Japan, September 2002.
6) R. Matsukura, Y. Karube, and T. Koyama: VizGrid: Virtualized Collaboration Environment with 3D Images. *FUJITSU Sci. Tech. J.*, **39**, 2, p.197-204 (2003).
7) Consortium of 3-D Image Business Promotion. (in Japanese). *http://rittaikyo.jp/*
8) 3D Consortium. *http://www.3dc.gr.jp/english/index.html*
9) M. Watanabe and T. Matsuzawa: Three-Dimensional Simulation of Flow in a Dissecting Aortic Aneurysm Reconstructed from CT images. International symposium on computational methods in biomechanics 2004, Madrid.
10) T. Matsuyama and T. Takai: Generation Visualization and Editing of 3D Video. Proc. 3D Data Processing Visualization and Transmission, Padova, Italy, June 2002, p.234-245.
11) N. Sakamoto, J. Nonaka, Y. Takai, K. Koyamada, and T. Matsuyama: View-Independent Texture Representation For Omnidirectional Display. Proceedings of the IASTED CGIM2003, 2003, p.135-140.
12) Yu Xiaowu: A research on streaming communication of volume data. Master Degree Thesis, Japan Advanced Institute of Science and Technology, 2004.
13) T. Tsuchiya, T. Komine, and Y. Tan: A Network Node for Synchronizing Multiple Streaming Data Based on Timestamp Information. Proceedings of The International Symposium on Towards Peta-Bit Ultra-Networks (PBit), A&I Ltd., ISBN4990033035, 2003, p.96-102.

Ryuichi Matsukura received the B.E. degree in Communication Engineering and the M.E. degree in Information Engineering from Tohoku University, Sendai, Japan in 1986 and 1988, respectively. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1988, where he has been engaged in research and development of mobile computers, an electronic meeting system, and remote collaboration systems. He is a member of the Association for Computer Machinery (ACM), the Information Processing Society of Japan (IPSJ), and the Japanese Society for Artificial Intelligence (JSAI). He received the Best Paper Award from the IPSJ in 2000.

Koji Koyamada received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from Kyoto University, Kyoto, Japan in 1983, 1985, and 1994, respectively. He is a professor at Kyoto University. From 1985 to 1998 he was a research member in IBM Japan. From 1998 to 2001 he was an associate professor at Iwate Prefectural University. From 2001 to 2003 he was an associate professor at Kyoto University. His research interests include scientific visualization and design optimization. He is a member of the IEEE Computer Society and the Information Processing Society of Japan (IPSJ). He received the IEMT/IMC Symposium Outstanding Paper Award in 1998.

Yasuo Tan received the Ph.D. in Computer Science from Tokyo Institute of Technology, Tokyo, Japan in 1993. Since then he has been working for Japan Advanced Institute of Science and Technology (JAIST). He is also an Associate Professor in the Information Network Chair of the School of Information Science at JAIST and an Associate Professor at the National Institute of Informatics (NII), Tokyo. His research interests include computer networks, especially home networking and ubiquitous computing. He is a member of the IPSJ, IEICE, IEEJ, IEEE, ACM, and Digital Home Networking Forum.

Yukihiro Karube received the B.E. degree in Industrial Chemistry from Nihon University, Tokyo, Japan in 1987. He joined Fujitsu Ltd., Tokyo, Japan in 1987, where he has been engaged in improvement of supercomputer performance and development and support of supercomputer systems.

Mitsuhiro Moriya received the B.E. degree in Applied Chemistry from Yamanashi University, Kofu, Japan in 1977. He joined Fujitsu Ltd., Tokyo, Japan in 1977, where he has been engaged in development of operating systems and support of computing systems for researchers and developers.