Annual Report of
ADVENTURE Project
ADV-99-1 (1999)

PARALLEL VISUALIZATION OF LARGE-SCALE FINITE ELEMENT SOLUTIONS USING PC CLUSTER IN THE CABIN

Izuru. SHIRAI[*1], Shinobu. YOSHIMURA[*1], and Genki. YAGAWA[*2]

*1 School of Frontier Science, University of Tokyo
e-mail: shirai@qs.t.u-tokyo.ac.jp, yoshi@q.t.u-tokyo.ac.jp

*2 School of Engineering, University of Tokyo

email: yagawa@q.t.u-tokyo.ac.jp

In dealing with large-scale finite element (FE) problems of over ten millions degrees of freedom (DOFs), it is very difficult not only to calculate but also to visualize them. This difficulty in visualization is mainly caused due to the following two reasons. Firstly, the technique for understanding large-scale analysis results that exceed our ability of perception must be different from usual techniques for small scale ones. Secondly, the visualization of large-scale analysis results, which are often solved using latest massively parallel processors (MPP), can not be handled on a conventional single processor machine any more. The authors are developing the visualization system for FE solutions using a virtual reality technique CABIN for the former issue, while adopting a PC-based cluster consisting of 24 PEs (DEC Alpha 533MHz) for the latter issue. CABIN provides a large scale and realistic space in immersive multiscreen display. This virtual reality technique makes it easy to understand the results, but require excessively computer resources. The PC-based cluster supplies the processed data to the front-ended CABIN system. Practical performances of the present system are demonstrated for some examples of fluid flow analysis.

Key words: Parallel Visualization, FEM, PC-based Cluster, CABIN,Virtual Reality

1 INTRODUCTION

Latest supercomputers make it possible to analyze large-scale problems of over ten millions degrees of freedom (DOFs). Scientific visualization plays a fundamental role in analyzing the calculation results, but current visualization techniques can not be matched to the large data set. Most visualization systems so far have a scale limit, and are devoted to a single powerful GWS such as SGI ONYX, because those techniques need global computing in the processes. This situation makes it difficult to visualize the results of ultra large-scale analyses. This is due principally to the following two reasons. Firstly, the technique for understanding those results that exceed our ability of perception must be different from normal techniques for small scale ones. Secondly, the visualization of those results that are often solved using latest massively parallel processors (MPP) can not be handled on a conventional single processor machine any more. The greater the scale of problems becomes, the more interactive data handling is important in visualization. This is because we perform such computational analyses to understand original complicated phenomena. So trial and error process is more necessary in this situation. For this interactivity, the present authors are developing the visualization system for large-scale problems using a virtual reality technique CABIN system. This technique, however, needs for more powerful computational resources, i.e. CPU, memory, disk and so on. For this reason, a PC-based cluster of 24 PEs (DEC Alpha 533MHz) is applied as a back-ended system. CABIN provides a large-scale and realistic space in immersive multiscreen display. A user of this system can understand an entire analysis result by linking displayed results simultaneously. The large data sets needed in this process are supplied from the back-ended PC-based cluster. Practical performances of the current visualization system are demonstrated for finite element solutions of fluid dynamics.

2 SYSTEM

2.1 PC-based Cluster

The PC-based cluster employed here is illustrated in Figure 1. This cluster is connected with 100Mbps FastEthernet among 24 PCs running Linux/Alpha using one switching hub. It also has an interface to the CABIN controlling workstation, i.e. SGI ONYX InfiniteReality with FDDI. Each PC has the large amount of memory of 768MB, and processes in parallel. Whole pre-processed data are finally concentrated to the CABIN system. This configuration is not always suitable for visualizing processes, because visualization sometimes needs not only local data handling but also global one. This process, however, should not be performed on the CABIN system because of excessive requirement for resources. The entire distilled data is concentrated in one processor before sending data to the CABIN system, and then this integrator processor extracts pre-final images as textures.

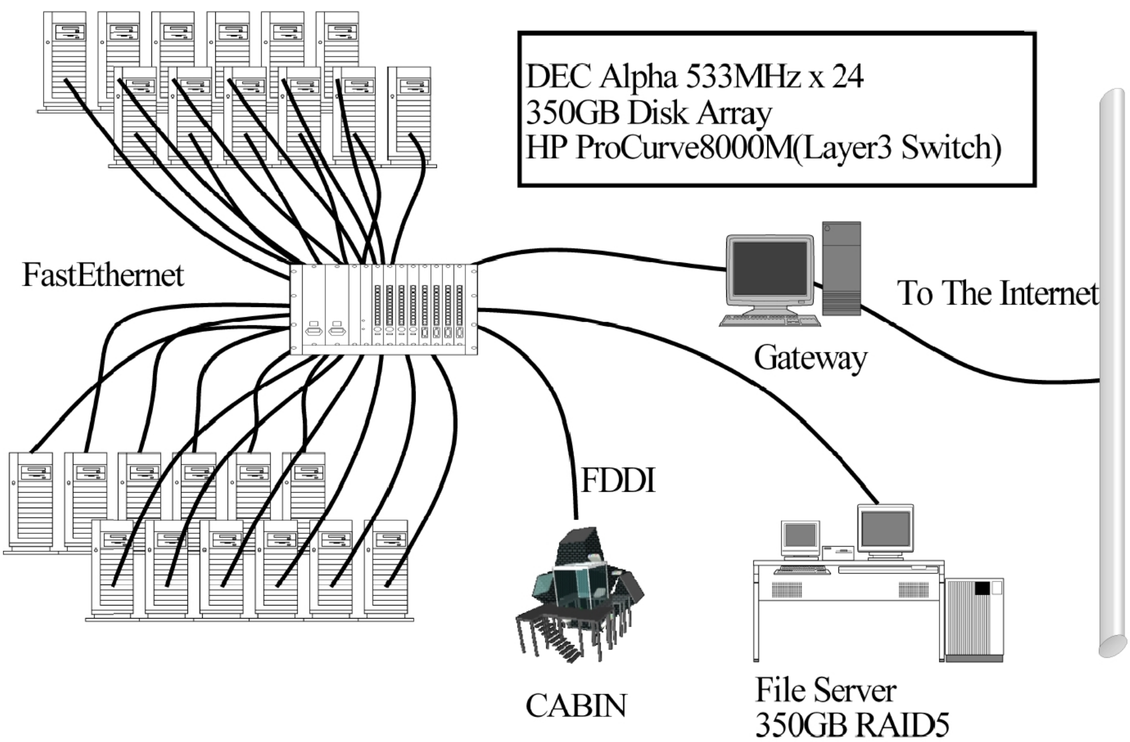


Figure 1 Alpha PC-based cluster and network

2.2 CABIN

Virtual reality environment called CABIN consists of 5 large screens (top, bottom, right, left and front) and 5 SGI ONYX InfiniteReality. The individual ONYX drives each screen, and a shared memory unit named ScramNet connects them. A display-synchronous shuttered glass is used for stereo-view. Their association makes an immersive multiscreen display. Figure 2 shows the exterior of CABIN.

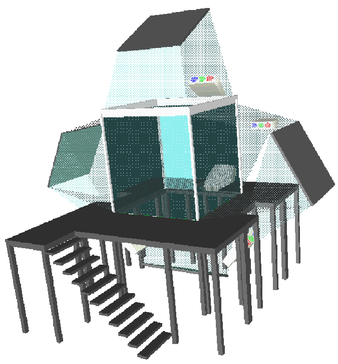


Figure 2 Exterior of CABIN

2.3 Data Flow Diagram

In virtual reality environment, the response time to users' input is one of the most critical issues. Then any processes should be shifted to the outside of the CABIN system as much as possible. For this requirement we propose the system shown in Figure 3. This model has the following three layers: the back-ended cluster, the front-ended CABIN system, and a mediator between two layers. The back-ended layer is the left-hand side surrounded by a circle in the figure, while the front-ended layer is the right-hand side. The mediator shown in the middle region, which is also the integrator of the parallel processing data, extracts textures and polygons. This is also the transport layer from front-ended to back-ended and vice versa.

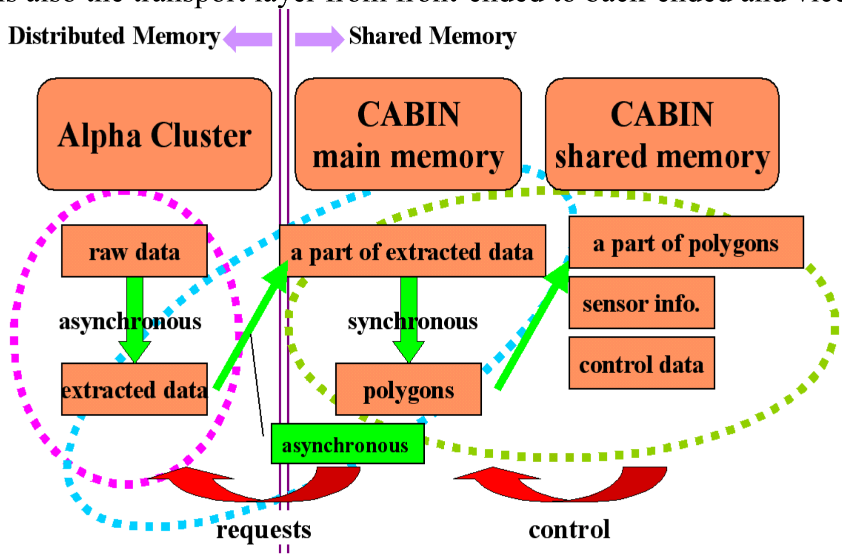


Figure 3 System data flow diagram

2.4 Implementation

In Figure 3 the synchronous communication is efficiently, because the processes of cluster and CABIN do not always finish at the same time. Especially the front-ended process must not delay because of the required frame rate in a virtual reality environment. To implement the system data flow diagram, we employed CORBA environment that makes event driven transaction possible. Figure 4 is a network map where the objects are connected using CORBA. Five objects around the virtual space mean five screens of CABIN. The numbers of objects at the left side are the results of analysis, and are caliculated asynchronously. The former objects and processes are the front-ended one, and the latter are the back-ended.

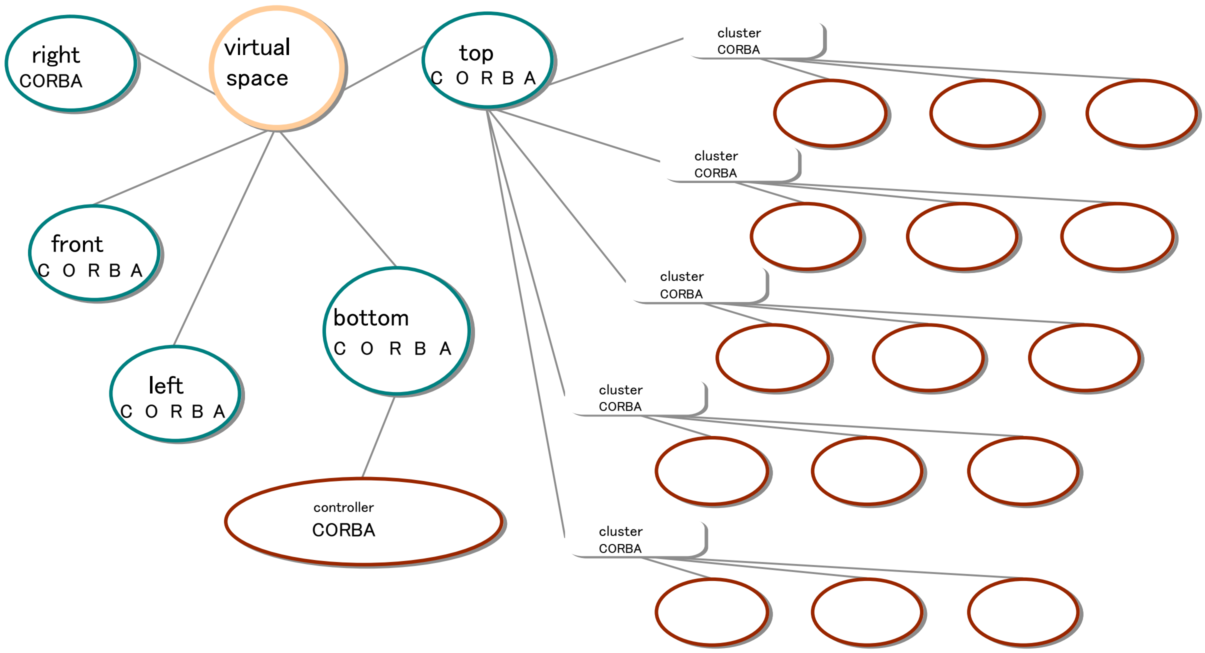


Figure 4 System implementation using CORBA

3 VISUALIZING PROCESSES

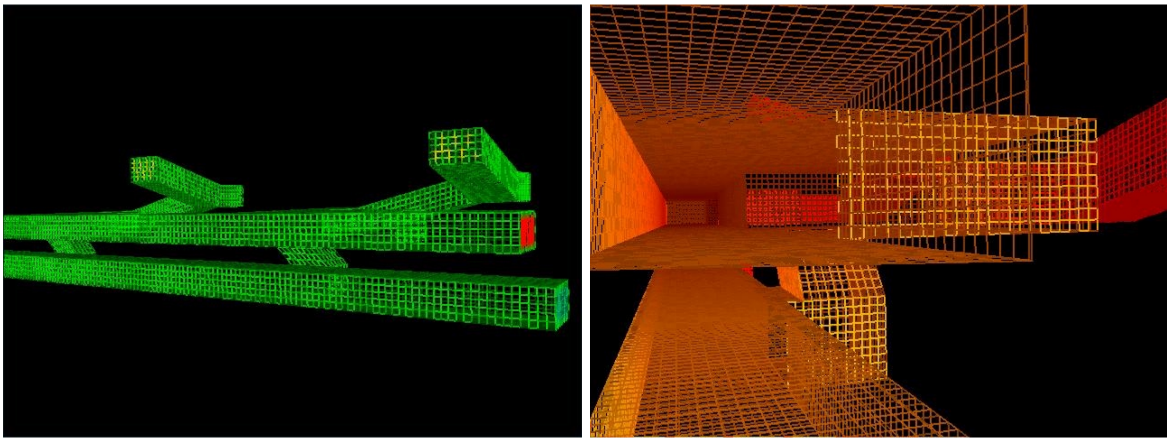
3.1 Progressive Resolution

The main targets of this system are finite element solutions. Firstly we deal with the results of fluid dynamics analyses as shown in Figure 4a. This figure displays pressure contours at the boundary surface of the analysis domain. The total number of polygons is about 16,000. In this case, practical performance of the CABIN system is satisfactory as a virtual reality environment. However if the problem scale becomes eight times as large as this example (see Figure 4b.), its response reduces less than 1Hz. Thus, we need to employ some kinds of simplification. It should be noted here that simplification of polygons is easy in this particular sample, but that our targets are more general analysis domains with complex shape. In the present study, we employ the progressive resolution control (PRC) in the middle layer of Figure 3 in order to resolve the problems. Taking the results shown in Figure 5 as an example, let us explain the PRC technique, which has the following 4 steps.

- draw a rough sketch using 22,000 DOFs solution

- detect a position of an operator, i.e. user, within the CABIN system
- extract the detailed data closest to the operator' position from 170,000 DOFs solution
- control the resolution if the preparation of data is in time

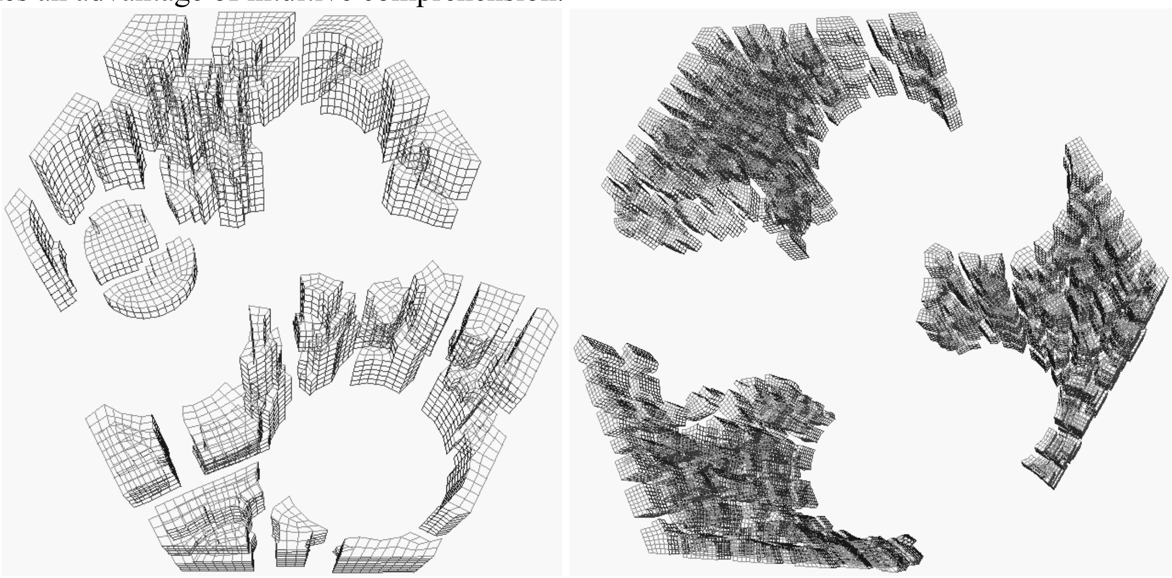
In consequence the operator can see the rough sketch interactively, and see the detailed data at his close position in a delayed response.



(a) **(b)**

Figure 4 Pressure contours at subway station

This PRC technique is effective in the following reason. Figure 5b shows a wire frame picture of a model decomposed into numerous domains, but no wire can be verified. This is because the raw data is too fine to identify. In such a case, drawing a rough sketch (Figure 5a) takes an advantage of intuitive comprehension.



(a) **(b)**

Figure 5 A model decomposed into several parts and numerous domains

3.2 Parallel Sampling of Cross Sections

Since a cross section is usually on one plane, only one polygon is necessary in a single processor environment. In a parallel environment, each processor draws a textured polygon one by one. At the mediator layer of Figure 3, these polygons are built up into one polygon. This process is illustrated in Figure 6. They are easily combined using opaque value (alpha value) when their shape is complex. Thus the PRC technique is to be applied in the same way as the surface rendering.

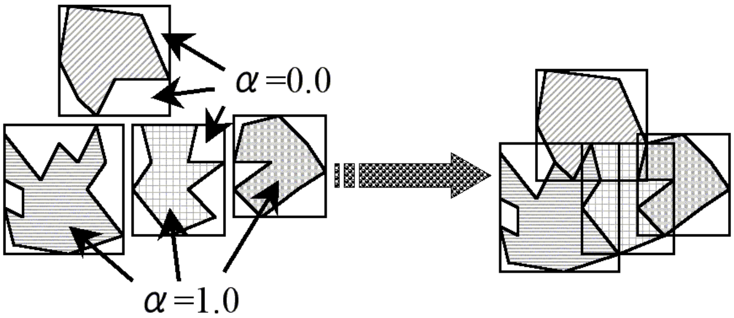


Figure 6 Assembling textures sent by each processor

4 DISCUSSION

4.1 Consideration of Warp Plain

Drawing cross sections seems easy at a glance, but is difficult for finite element solutions. It is not always true that elements are composed of simple plains. For example, the nodes of a cube element are not always sitting in one plane. It means that the deterministic extraction of cross sections is not possible using the node connectivity of each element. In other words, Marching-Cube or its derivative methods are not suitable in this case. For resolving this difficulty, the system computes the value at the exact plain of cross section after influential elements are decided. Figure 7 shows the diagram.

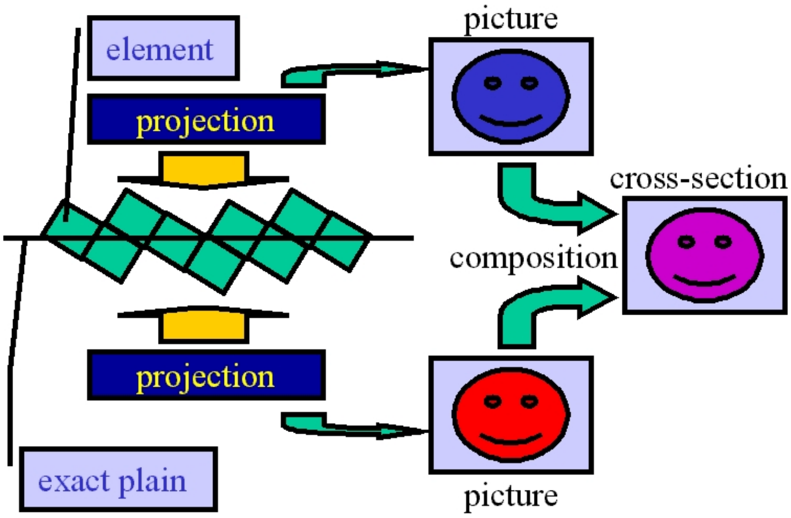


Figure 7 Diagram for extracting textures of cross sections

4.2 Evaluation of Computing Time and Memory Usage

We evaluate the requirement in resources computing the textures. Here we examine computing time and memory usage for the following three items.

- extracting boundary surfaces
- extracting textures of a cross section
- total frame-rate in viewing

Table 1 shows the results. This results show that the developed progressive resolution control method is necessary for the CABIN system. It should be noted here that the rough sketch (22,000DOFs) operates interactively and its frame-rate is 15Hz. If the user take notice of one scene, the detail (170,000DOFs) will come out within a second. In dealing with 1,250,000 DOFs problems, the requirement of memory usage is 487 Mbyte. If this were a limit on one processor, the system world have the capacity for visualizing 30 millions DOFs' solution using 24 PCs. Of course, this estimate is in case that the back-ended cluster is not applied to the CABIN system but to CRT-based system.

Table 1 Computing time and memory usage

DOFs	surfaces	cross section	frame-rate	memory usage
22,000	< 1 sec	< 0.1 sec	15 Hz	10 MB
170,000	1 sec	1 sec	1.0 Hz	71 MB
1,250,000	5 sec	2 sec	0.2 Hz	487 MB

4.3 Isosurfaces

Ordinary processes for isosurface consist of the following three steps.

- retrieve the MAX and MIN values among the node values on elements
- extract some elements, which satisfy the equation (MIN $\leqq$ threshold value $\leqq$ MAX)
- calculate surfaces from the elements

Contortions of elements obstruct the deterministic methods using element topology. When processing larger scale problems, the number of surfaces increases, and then the mediator processor is overloaded. Therefore we employ the algorithm as shown in Figure 8. Firstly, series of images of a cross-section are calculated, and then are contoured on the back-ended processors. Finally, the master PE builds isosurfaces. The resolution is determined by the interval of cross-section employed.

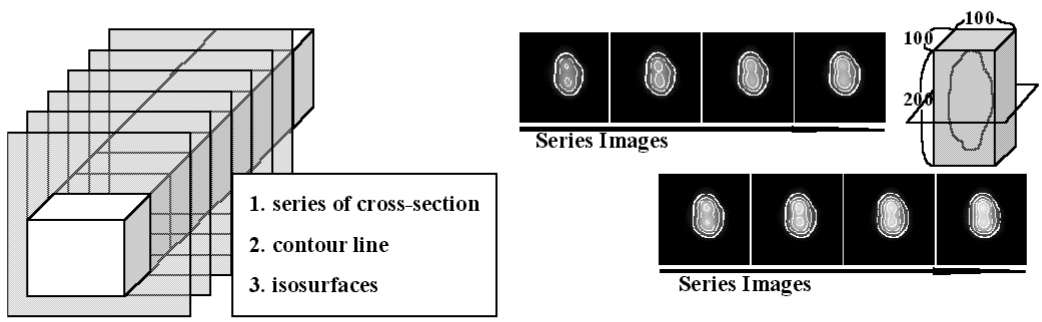


Figure 8 One processor builds isosurfaces from series of cross-sections

5 CONCLUSION

We developed an interactive parallel visualization system for large-scale finite element solutions in a virtual reality environment (CABIN). The CABIN system requires small latency in rendering frames. For this reason we employed PC-based cluster and parallel techniques. To determine practical performance, some finite element solutions are visualized with the system. The satisfactory result is obtained for time of extracting surfaces needed for the CABIN environment. The system of left part in Figure 3 can be used not only for the CABIN system but also for a normal CRT-based visualization. In this case it enables to visualize ultra large-scale solutions (100M DOFs).

REFERENCES

[1] Annual Report of ADVENTURE Project (Aug. 1, 1997 – March 31, 1998), ADV-98-1, (1998)
[2] Michitaka Hirose (1997), Development of an Immersive Multiscreen Display (CABIN) at the University of Tokyo, Proc. 1st International Immersive Projection Technology Workshop, pp.67-76
[3] http://adventure.me.tokushima-u.ac.jp/
[4] http://adventure.q.t.u-tokyo.ac.jp/