1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

DISTRIBUTED VIRTUAL ENVIRONMENT FOR VOLUME BASED SURGICAL SIMULATION VIA THE WORLD WIDE WEB

[a]*Ming-Dar Tsai,* [b]*Ming-Shium Hieh* *and* [c]*Wen-Chien Chang*

[a]Institute of Information and Computer Engineering,
Chung Yuan Christian University, Chung-Li, Taiwan/R.O.C.
Email:tsai@iccad.ice.cycu.edu.tw

[b]Department of Orthopaedics and Traumatology,
Taipei Medical College Hospital, Taipei, Taiwan/R.O.C.
Email:shiemin@mail.tmc.edu.tw

[c]Oral and maxillofacial Division, Department of Dental,
Taipei Medical College Hospital, Taipei, Taiwan/R.O.C.
Email:wenchein@mail.tmc.edu.tw

ABSTRACT

It needs long time to educate and train a surgeon because he has few opportunities to rehearsal his learned surgical modalities on patients. To solve the problem, we propose a low-cost distributed VR architecture via the WWW. In this architecture, Web pages from the server drive VR I/O devices on clients to let users immerse in virtual environment. This is different from the conventional approach that the server is responsible for most works and devices. In this paper, we focus on the application of volume based surgical simulation that is usually considered consuming large computation and acquiring high-priced workstations. We built a system that uses PC platforms and the WWW media to accomplish the distributed VR surgical simulation. We introduce the simulation algorithms and the methods for cooperating the clients with the server. Finally, we show an example of muskuloskeletal surgery.

1. INTRODUCTION

Today, Web browsers have been ported on most computer platforms for reading the HTML files [1]. Rapidly growing numbers of servers broadcast their works as hypermedia documents to allow low cost public access through the World Wide Web (WWW). Instead of only text or static graphics, some server sites begin to enrich their Web pages by using Java or VRML to add user interactions, describe 3D scenes and object behavior [2-3]. The medium of the WWW no doubt offers great opportunities in many applications. This paper shows a possibility of using this medium to simulate volume based muskuloskeletal surgery under virtual environment (VE).

A surgical simulation system takes physical data from individual patient to make a simulation that will help plan and rehearse surgical procedures both for verifying and teaching procedures of an operation. Simulation systems combining virtual reality technology can provide users computer-based generation of 3D visual, and even auditory or tactile environments and 3D input tools that allow surgeons to immerse, navigate and interact with virtual patients. Virtual reality (VR) simulation can duplicate the operation field and thereby enhance training and reduce the need for expensive animal training model [4]. However, in order to produce a sense of realism for medical purposes, developers bare faced with providing organ fidelity, interactivity, physical and physiological properties of organs and sensory input. However, the integrated package requires large amounts of computational power, which has traditionally come from mainframes (e.g., [5-9]). This is considered as a drawback to promote VR system to practical uses. But the economic barrier to VR's uses in medicine will be much lessened because of the price dropping in CPU, RAN and 3D graphics accelerator. VR techniques begin to work on products based on lower-cost desktop systems. For example, London-based Virtual Presence's MIST system that teaches standard psychomotor skills for laparascopic surgery is based on a Pentium 133 and sells for around $15,000 [10].

The hardware and software of Tele-surgery system are more expensive and special. For examples, Satava developed a prototype system of using remote sensors to determine how severe soldiers wounded then to provide teleconsultation and tele-presence surgery [11]. Arai et al. reported a real-time intravascular tele-surgery based on multimedia communication by using high-speed optical fiber to connect two places with 350-km distance [12].

The objective of this research is using the HTML language to provide a "point-and-click" interface for simulating surgery under VE through the WWW. The scenario is that the server provides software packages on the WWW to simulate surgery and to drive 3D I/O devices such as shutter glass, head mount display, glove and tracker equipped on clients. In this research, the simulator on the server manipulates medical volumes and reconstructs triangulated isosurfaces from the volumes. While, PC clients are responsible for rendering computation to generate 3D shaded images and stereographic images of different perspectives. The clients can be equipped with a shutter glass to observe the stereographic images and a tracker to simulate the surgical procedures.

Currently, the prototype system is used to educate and train residents of orthopedic department of Taipei Medical College Hospital. Trainers can operate some different surgical modalities to show their effects for correcting the same deformity. They can get more real ideas about the whole operation and every procedure of a surgical modality trough the visual assistance of VR. Visiting doctors also use the system to rehearse surgical modalities before operation, confirm and discuss surgical plans, and may try new modalities. Because of using PC platforms and the WWW media that are familiar to most people, it is easy to use and not expensive to set up this kind of client.

2. SYSTEM ARCHITECTURE

Figure 1 shows the system architecture. The system includes four modules at a client. The interface module and the socket module are implemented by a Java applet. The interface module interprets and processes every command from the user. However, the hardware on the client can not be accessed by the Java applet. Therefore, We packed the drivers of the hardware as ActiveX controls that follow the protocol of COM (Component Object Model) [13]; then, use the VBscript language to deliver data between the ActiveX controls and the Java applet.

When a client downloaded our Web page from the Internet, the Java applet of the interface module and the socket module starts. The VBscript starts the ActiveX controls to check whether 3D I/O devices on the client are acceptable and initialize these devices if acceptable. The socket module creates a socket to connect with the server. The interface module provides panels to let the user input various parameters and commands. The rendering module is an ActiveX control that uses the OpenGL library to drive the 3D graphics accelerator. The triangulated isosurfaces from the server are rendered at this module. The tracker module is also an ActiveX control that accepts the data about the positions and angles of the tracker; then, sent the data to the server for implementing surgery simulation. Section 3 introduces the interface, rendering and tracker modules.

The system includes four modules on the server. The triangulation module reconstructs triangulated isosurfaces from medical volumes by using the marching cube algorithm [14]. In this module, a cuboid volume can be divided into several smaller cuboid subvolumes as the user requested. This module independently reconstructs isosurfaces of each subvolume. Because most part of a medical

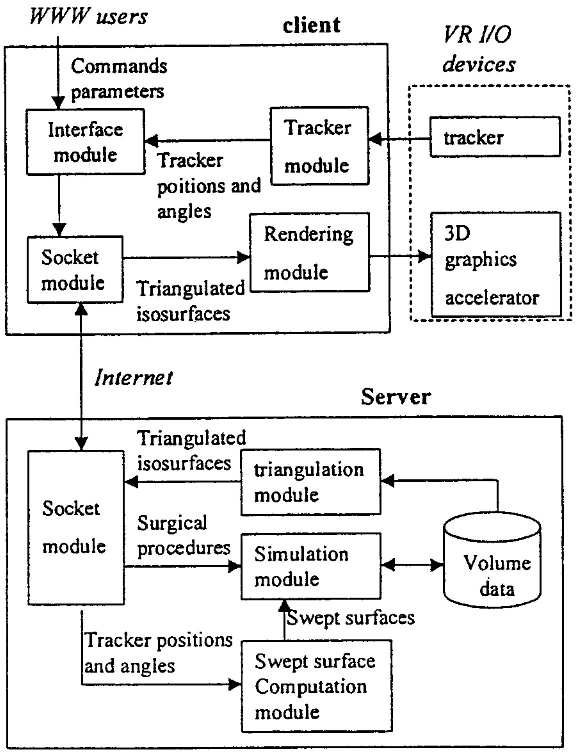


Figure 1 System architecture

volume is not changed during simulation, the server does not need to reconstruct the triangles of the whole volume. We can improve efficiency if only reconstructing the triangles of the changed subvolumes during simulation. A user can divide a volume as a non-operated subvolume, and some independent operated subvolumes. The non-operated part is considered as not changed during simulation. Only one subvolume will be operated during the simulation of one surgical modality. For example, a skull for orthogonathic surgery simulation can be divided into the brain (non-operated part), mandible or maxilla (independent operated parts). Surgeons usually operate only on the maxilla or the mandible under one surgical modality.

The server provides the functions of manipulating volume data to simulate the corrective osteotomy of the muskuloskeletal system that is used to correct deformities of bones and joints. Surgeons use the osteotomy to solve functional and aesthetic problems by skeletal deformities. We modified the simulation algorithms as reported on [15]. Surgeons use the following function to operate on computer-generated patients as the actual surgical procedures: sectioning anatomic structures of bone and holding (recognizing) separate anatomic structures, removing, translating and rotating separate structures, fusing and healing up separate structures and associated soft tissues. To implement under VE, we modified the simulation system as in a 3D I/O fashion. The user can use surgical instruments to operate on stereographic images. The swept surface computation module computes swept surfaces of the instrument based on the positions and angles of the tracker attached on the instrument. Section 4 introduces the swept surface computation and the simulation algorithms.

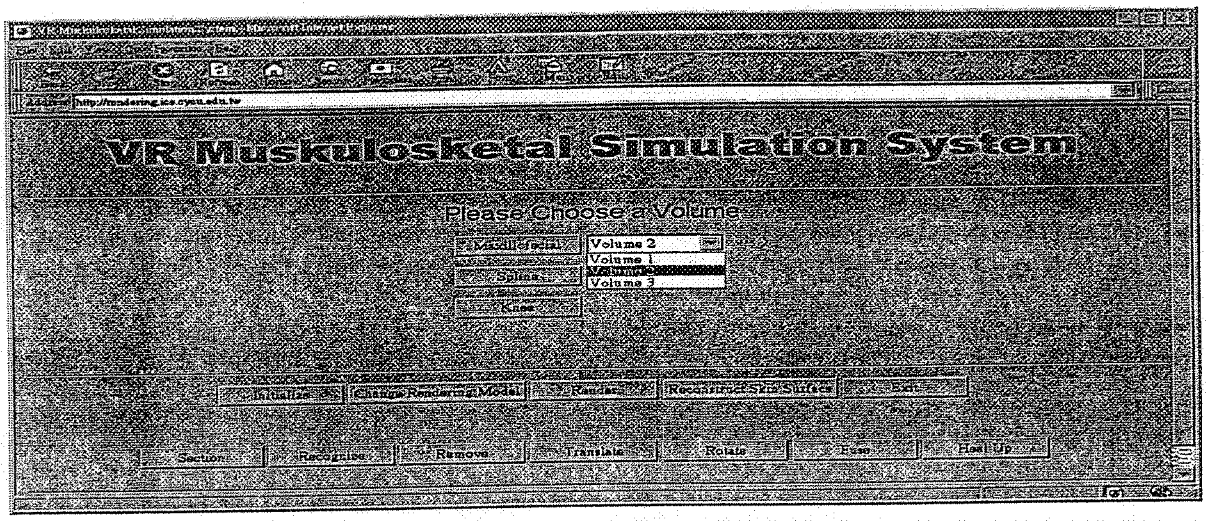
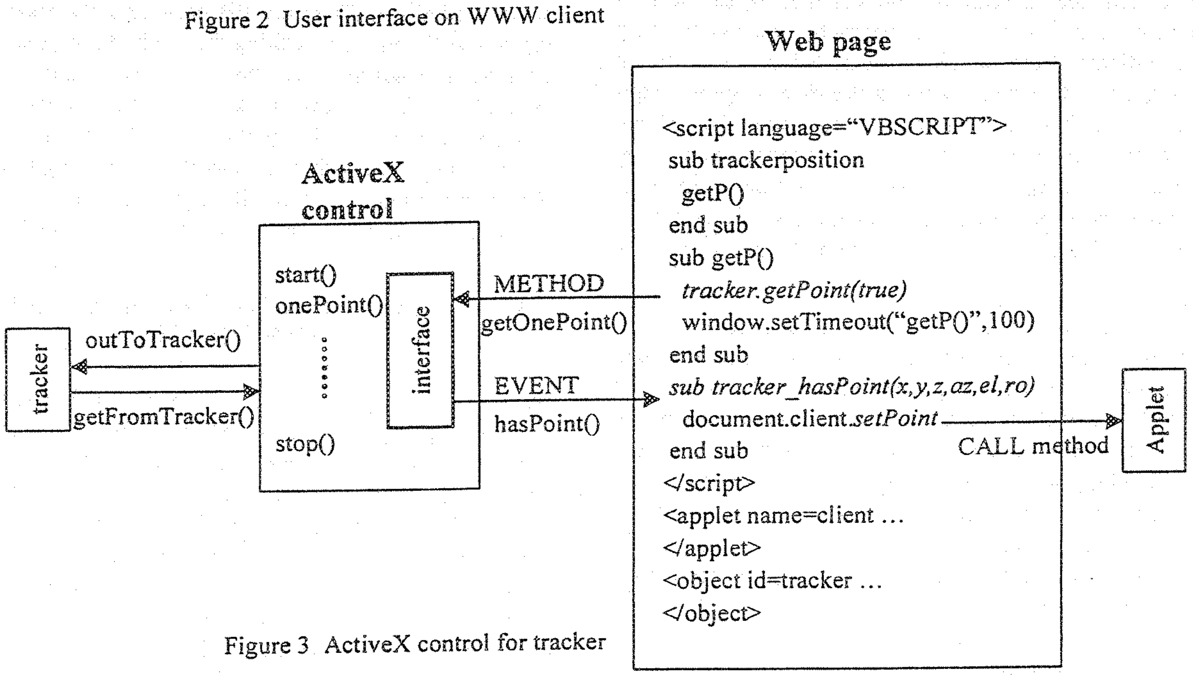
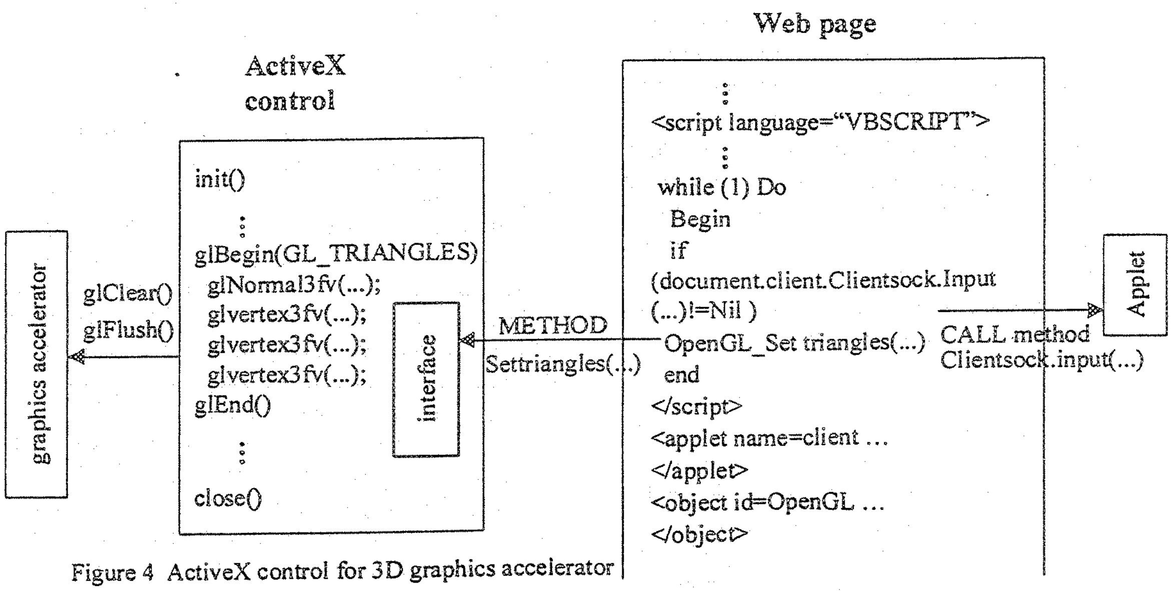


Figure 2 User interface on WWW client



```
<script language="VBSCRIPT">
 sub trackerposition
   getP()
 end sub
 sub getP()
   tracker.getPoint(true)
   window.setTimeout("getP()",100)
 end sub
 sub tracker_hasPoint(x,y,z,az,el,ro)
   document.client.setPoint
 end sub
</script>
<applet name=client ...
</applet>
<object id=tracker ...
</object>
```

Figure 3 ActiveX control for tracker



```
init()
  ⋮
glBegin(GL_TRIANGLES)
  glNormal3fv(...);
  glvertex3fv(...);
  glvertex3fv(...);
  glvertex3fv(...);
glEnd()
  ⋮
close()
```

```
      ⋮
<script language="VBSCRIPT">
      ⋮
 while (1) Do
   Begin
   if
(document.client.Clientsock.Input
(...)!=Nil )
   OpenGL_Set triangles(...)
   end
</script>
<applet name=client ...
</applet>
<object id=OpenGL ...
</object>
```

Figure 4 ActiveX control for 3D graphics accelerator

3. VR DEVICE DRIVERS ON CLIENT

3.1 Interface Module

The interface module is actually a Java applet. The applet shows an interface panel as illustrated in Figure 2. By pointing and clicking on the panel, the user can send one of the following commands: choosing a volume, initializing this volume, setting parameters of the rendering model, render the volume, simulating surgical procedures (section, recognize, remove, translate, rotate, fuse and heal up) by the tracker.

When a volume is chosen, the server sends a pre-rendered stereographic image of the volume to let the client show it on an applet frame. The stereographic image also shows the origin and three primary axes of the volume. After a user determined to initialize the volume, he can wear a shutter glass to observe the stereographic image and to set the coordinate of the volume by using the tracker to match the origin and the primary axes on the stereographic image. Then, the module requests the user to divide the volume and send the division information to the server. The server reconstructs triangulated isosurfaces of bones for each subvolume and sends them to the client to shade. However, the server also reconstructs isosurfaces of the skin if the user requests.

After initializing a volume, the user can choose one of the surgical commands (section, recognize, remove, translate and rotate, fuse and heal up) to operate on the volume by the tracker. The client sends the positions and angles of the tracker and the command type to let the server simulate a corresponding surgical procedure. The user can change parameters of rendering models including materials, lights, viewing conditions and the shading model (Phong model). He can also change these parameters during implementing the rendering program.

3.2 Tracker Module

Figure 4 shows the ActiveX control for tracker and how it communicates with the Java applet (interface module). The control provides a method that the Web page by the VBscript language use it to request the data of one tracker point. A tracker point includes six floating numbers, three coordinates and three angles. When the Web page requests, the control will send the data of the tracker point to the applet as an event. The Java applet send the data to the server with one of the following commands including initializing a volume and simulating on the volume (section, recognize, remove, translate and rotate, fuse and heal up).

Currently, our system provides the ActiveX control for the InsideTRAK tracker (made by Polhemus Inc.). The tracker is actually a transmitter that generates a magnetic field to detect the coordinate and angular attitude of a receiver attached on a surgical instrument. Through a specific I/O port, the control use the functions of "start", "stop" and "onepoint" to initialize, close and request the tracker. Then, the tracker sends raw data about positions and angular attitudes of the receiver through another specific I/O port. The control interprets the raw data as a tracker point including six floating numbers, three coordinates and three angles. Although this tracker can provide data of continuous points of the receiver, we let the control accept the data as discrete points under at least 10ms intervals. Under the same specific I/O ports, the ActiveX control of the tracker can work for any PC platform.

3.3 Rendering Module

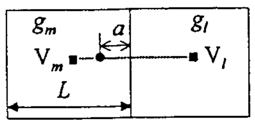
Figure 4 shows the ActiveX control for 3D graphics accelerator and how it communicates with the Java applet. The Java applet provides a method to let the VBscript accept triangulated isosurfaces from the socket connecting the server and client. Then, the VBscript send the triangles to the ActiveX control by calling the method provided by the control. The control uses the OpenGL to drive the 3D accelerator for rendering the triangles. Because the OpenGL is standard library, the ActiveX control should be valid for any 3D graphics accelerator supporting OpenGL. We have verified the ActiveX control for the accelerators of the Gloria XXL, Gloria L, Gloria M (by ELSA Inc.).

4 MUSKULOSKELETAL SIMULATOR ON SERVER

For manipulating volume data to simulate the muskuloskeletal surgery, we use boundary pointers to represent and simulate boundary changes of bone structures and soft tissue; then, normalize the values of voxels to let the values not depend on tissue types. Because the voxel values are normalized, reposition of bone structures or soft tissues does not influence the values of their surrounding voxels. Therefore, we can only change contents of voxels to simulate various surgical procedures including sectioning, recognizing, removing, translating, rotating, fusing and healing up anatomic structures and associated soft tissue.

4.1 Data structure of volumes for surgical simulation

We define the column direction of a volume as *Z*-axis direction, the row direction as *Y*-axis direction. Tomographic slices are put sequentially along *X*-axis (slice) direction. Each voxel is given two boundary pointers pointing to the next boundary voxel in *Z*-axis direction and -*Z*-axis direction. We also use the equation as shown in Figure 5 to normalize the values of boundary voxels. The voxels V*m* and V*l* belong to different types of tissues; therefore, there exists a boundary between them. V*m* represents a voxel of more important tissue. V*l* represents a voxel of less important tissue. The importance priorities for tissues are bone, other soft tissues, skin then air. *L* is the voxel length. *sm* and *sl* are the original values of V*m* and V*l*. The threshold *T* represents the value of the boundary. *gm* and *gl* are the normalized gray-level values of V*m*



$$\frac{g_m \times L}{N} = a = \left(\frac{S_m - T}{S_m - S_l} - \frac{1}{2} \right) \times L$$

Figure 5 Normalization

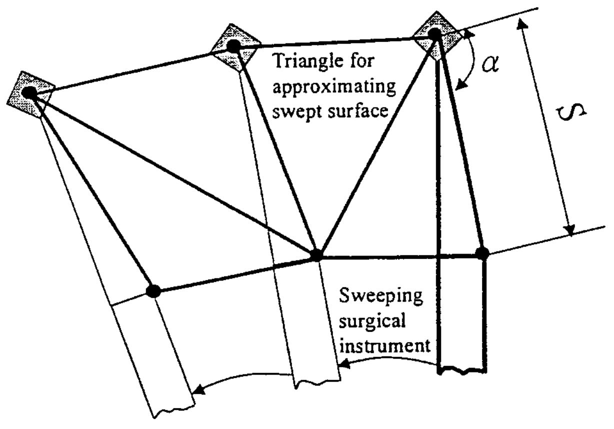

Triangle for approximating swept surface
α
S
Sweeping surgical instrument

Figure 6 Computation of swept surface of surgical instrument

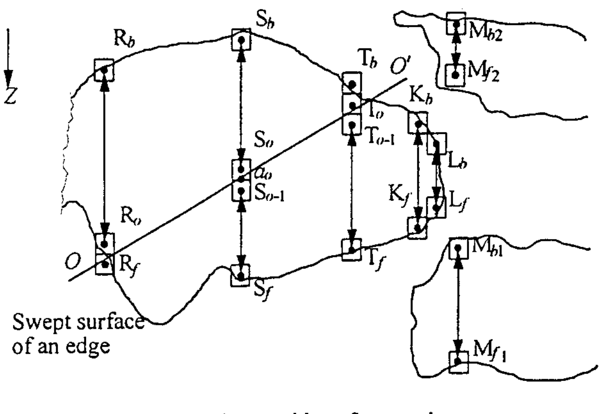


Figure 7 Section and recognition of anatomic structure

and V*l*. *N* (62) is the number of gray levels. The sample point *a* represents the boundary, it is an intersection of the boundary with the line connecting the centers of V*m* and V*l*. We compute *a* from *sm* and *sl*, *T* and *L*. Because *a* will not change after normalization, we use it to determine *gm*.

4.2 Computation of swept surface

We compute the position of surgical instrument by the information of only one tracker. As illustrated in Figure 6, the tracker is attached on one end of the instrument. The other end of the instrument is computed by the position of the tracker, α the angular attitude of the tracker and *S* the length of the instrument.

We use triangles to approximate the swept surface of the instrument. As the example shown in Figure 6, we can obtain two approximate triangles from two positions of the instrument by connecting the tracker end of the next position with the opposite end of the current position.

4.3 Section of anatomic structure

For each triangle swept surface, we compute the (sample) points that the triangulated swept surface intersects with the lines connecting the voxel centers along the Z-axis direction. As a simplified 2D example shown in Figure 7, an intersection point *ao* locate inside the pair of boundary voxels S*f* and S*b*. Two new boundary voxels S*o*-1 and S*o* are generated by the swept surface. We change the boundary pointers of S*f*, S*b*, S*o*-1and S*o*, and normalized values of S*o*-1 and S*o* to represent the section. The boundary pointer along -*Z*- axis direction of S*f* is changed as *z* coordinate of S*o*-1. The pointer along *Z*-axis direction of S*b* is changed as *z* coordinate of S*o*. For the new boundary voxel S*o*, we give its boundary pointer along -*Z*-axis direction *z* coordinate of S*o*-1, the pointer along *Z*-axis direction *z* coordinate of S*b*. For S*o*-1, we give its pointer along -*Z*-axis direction *z* coordinate of S*o*, the pointer along *Z*-axis direction *z* coordinate of S*f*. Using *ao* we can compute the values of and S*o*-1and S*o* by the equation shown in Figure 6.

4.4 Recognition of anatomic structure

Recognizing an anatomic structure means to find all boundary voxels belonging to the structure. Because boundary pointers record the boundary information in *Z*- or -*Z*-axis direction, finding the boundaries of an anatomic structure can be considered as a 2D problem. That means we can recognize a structure if we know the endmost columns of the structure. The endmost columns show the extents of the boundary voxels on *XY*-plane. We use the seed and flood algorithm to find the boundary voxels inside the endmost columns. The seed and flood algorithm is a recursive technique used to fill an area where 2D boundaries (endmost columns) have been drawn closed. Here, the closed boundaries are composed of sectioned boundaries and natural boundaries. For obtaining a sectioned boundary, the surgeon is requested to give each side of the swept surface a different structure code during sectioning. The two structure codes are used to represent the right and left divided structures. As the example shown in Figure 7, the pair of the boundary voxels T*f* and T*b* from where the edge left the sectioned structure belongs to *T* an endmost column of the left structure. Its structure code is given the one as to the right side of the swept surface. The boundary voxel R*f* from where the edge entered the sectioned structure belongs to *R* an endmost column of the right divided structure, the structure codes is given the one as to the left side of the swept surface. Natural boundaries are determined by the extents of a pair and its neighboring pair of boundary voxels. If parts of their extents overlap (as *K* column in Figure 7), we consider they belong to the same structure. Otherwise, they belong to different struc-

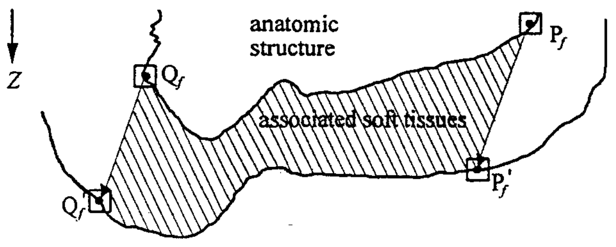


Figure 8 Determination of translating volume of soft tissues

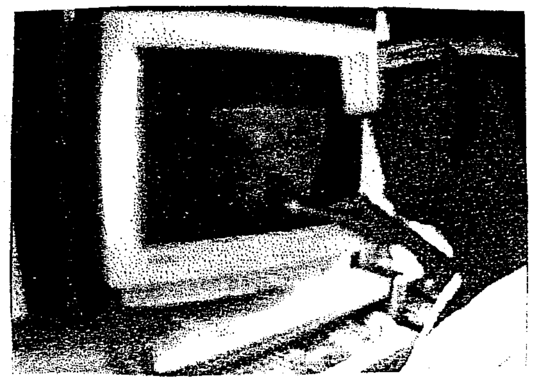


Figure 9 Implementing surgery simulation under virtual environment with a shutter glass and a tracker

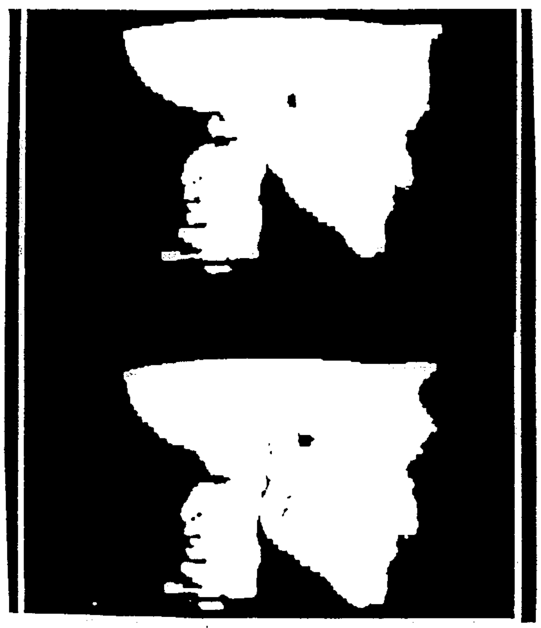


Figure 10 A pair of images for stereogarphics

tures (as *L* column in Figure 7) and a natural boundary exists.

When the surgeon held a sectioned structure to move, he specified a seed voxel on the structure for recognizing and gave it a structure code. The seed voxel and its corresponding boundary voxel are a recognized pair (seed pair), and the pairs of boundary voxels on the endmost columns by sectioned boundaries are also recognized. Then, the system begins the search at the four surrounding pairs of the seed pair and repeats the search at the four surrounding pairs of recognized pair. If the neighboring pair of boundary voxels is already recognized or not belonging to the structure, we stop the recursion to the neighbors as the seed and flood algorithm. As the result, if the endmost columns of the structure form a closed area, the recognition computation will stop after all pairs of boundary voxels inside the structure are recognized.

4.5 Translation of separate structure and associated soft tissues

Surgeons can operate separate structures with the following functions: removing, translating, rotating, fusing and healing up. Here, we introduce the algorithm of the translating function. For translating a separate structure, we clear the voxels where the structure was, then write the structure to the new place. We change boundary voxels of the structures as internal voxels to clear the structure. To change a boundary voxel as an internal voxel, boundary voxels that the boundary pointers of the voxel pointing to will become pointing to each other. To translate the structure, we write the contents of the boundary voxels to the voxels of the new place where the structure is translated to. However, the boundary pointers are added z component of the translation vector.

Simulating soft tissue changes is actually complicated because soft tissues are not rigid. Usually, finite element method (FEM) must be applied for computing a rather precise result (e.g., [16]). However, it becomes computation expensive. Because the reposition of a structure in the corrective osteotomy is usually small, we simplify the reposition of associated soft tissues as rigid and deal with associated soft tissues of a structure as a separate volume but just conjoined to the structure. We use the translation vector and the boundary voxels on the endmost columns of the translating structure to compute the separate volume of its associated soft tissues. As the example shown in Figure 8, the translation vector starting from a boundary voxel of the structure to the surface of the soft tissues forms a section surface. After processing the section and recognition computation, we can obtain a separate volume of soft tissues. A translation simulation is implemented by translating the separate volume of the associated soft tissues and the separate structure.

5 IMPLEMTATION EXAMPLE

In our prototype system, the server uses Pentium-II 400 and 256M RAN. One PC client use Pentium-II 233, with a 21" monitor, 64M RAN, a 3D graphics accelerator (Gloria XXL by ELAS inc.) and also equipped with a shutter glass (CrystalEyes PC by StereoGraphics) and a tracker (InsideTRAK by Polhemus). It takes near $4,000 to set up the client.

Figure 9 shows a resident uses a surgical instrument attached with a tracker to match the origin and the axes of a volume constituted by a $256 \times 256 \times 45$ CT data of a head. Figure 10 shows the pre-rendered images that form the

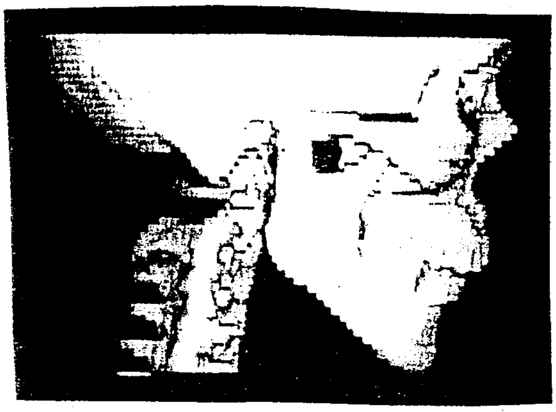


Figure 11 Lateral view before simulation

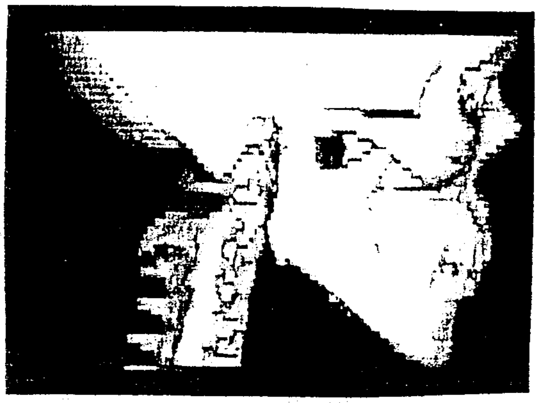


Figure 13 Lateral view after simulation

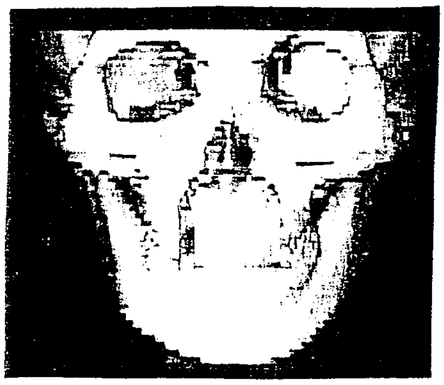


Figure 12 Removing the right premolar tooth, and sectioning the left premolar tooth

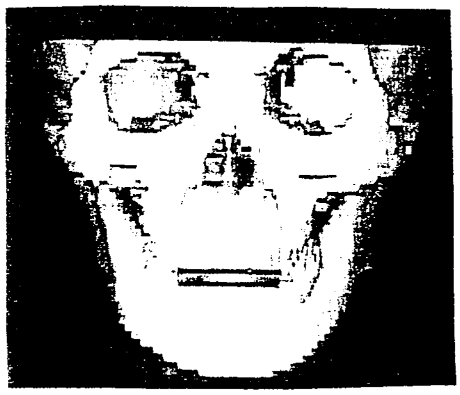


Figure 14 Frontal view after simulation

stereographic image as shown in Figure 9. The two images with the origin and primary axes of the volume are rendered under the same parameter, just one for left eye (upper part), the other one for right eye (lower part). After initialized the volume, the server reconstructs triangulated isosurfaces of bones and sends them to the client. It takes 40 seconds to reconstruct 338,156 triangles. At the client, it takes about 1.5 seconds to obtain a stereographic image.

Figure 11 shows a rendering result of the skull. Because the mandible belongs to a type of retrusion, the surgeon uses the modality of the upper anterior subapical osteotomy to move the maxilla backward. In the simulation for this modality, the server only reconstructs isosufaces of the subvolume of the maxilla that is about 1/200 of the whole volume. Figure 12 shows the surgeon sectioned and removed the right upper premolar teeth sockets, then sections the left premolar tooth and socket. The section and removal computation for a premolar tooth and socket take near 1 second. Figure 13 and 14 show the lateral and frontal views of the rendering results after fusing the skull and the translated maxilla that takes near 2 seconds to implement the simulation. Comparing to the rendering result before simulation as illustrated in Figure 11, the simulation results show the above modality can correct the deformity of the mandible retrusion.

6 Discussion and Future Works

We proposed a distributed VR surgical simulation system via the WWW. Different from the usual WWW applications, our system utilize the VR I/O devices on clients to let WWW users implement surgery simulation under virtual environment. Although the transmission time and isosurface reconstruction time for a whole volume can not offer time-critical interaction, the user can still obtain fast interactions of observing different perspectives of a volume by the VR I/O processing ability of the client. Although the prototype system is now experimented in a small group of Taipei Medical College Hospital, we view this research as a feasibility study of new applications using the Internet for widespread uses. Remote medical students and residents might access the VR resources to experience surgical modalities on textbooks or reports

Future works include the following three aspects. First, we should add a haptic function to the system by using force feedback devices on clients to let users obtain more realistic feeling when sectioning or repositioning bone structures. Second, we have to improve the server performance by distributed computing if we want to let public users access our web site. Third, we can employ the hypermedia inheritance of the WWW to build a computer-aided instruction system that assist surgeons to diagnosis, treat, analysis, plan and rehearse surgical cases.

ACKNOWLEDGMENT

This research work was sponsored by National Science Council, a grant number NSC-87-2213-E033-005. This support is gratefully acknowledged. We also grateful to the visiting doctors and residents of orthopedic Departments of Taipei Medical College Hospital for testing the system and their helpful suggestions

REFERENCE

[1] Berners-Lee, T., Cailliau, R., Groff, J.F. and Pollerman, B., “World-wide web: The information universe”, *Electronic Networking: Research, Application and Policy*, 1(2), Westport CT, Spring, 1992.

[2] Chan, P., “The Java Class Libraries : an Annotated Reference”, *Addision-Wesley*, 1996.

[3] Hardman, J. and Wernecke, J., “The VRML 2.0 Handbook, Building Moving Worlds on the Web”, *Addision-Wesley*, 1996.

[4] Ota, D., Loftin, B., Saito, Tim., Lea, R. and Keller, J., " Virtual Reality in Surgical Education", *Computer in Biomedicine*, 25(2), pp.127-137, 1995.

[5] Bainville, E., Chaffanjon, P. and Cinquin, P., " Computer Generated Visual Assistance During Retroperitonenscopy", *Computer in Biomedicine*, 25(2) , pp. 165-171, 1995.

[6] Hunter, I.W., Jones, L.A., Sagar, M.A., Lafontaine, S.R. and Hunter. P.J., " Ophthalmic Microsurgeical Robot and Associated Virtual Environment", *Computer in Biomedicine*, 25(2), pp. 173-182, 1995.

[7] Ziegler, R., Fischer, G., Muller, W. and Gobel, M., " Virtual Reality Arthroscopy Training Simulator", *Computer in Biomedicine*, 25(2),Vol. 25, No. 2, pp. 193-203, 1995.

[8] Kuhlen, T. and Dohle, C., " Virtual Reality for Physically Disabled People", *Computer in Biomedicine*, 25(2), pp. 205-211, 1995.

[9] Hong, L., Muraki, S., Kaufman, A., Bartz, D. and He, T., “Virtual Voyage: Interactive Navigation in the Human Colon”, ACM Computer Graphics, 31(5), pp.27-34, 1997.

[10] “VR in Medicine – Part 1: Current Developments”, VR news, 5(7), pp.24-27, 1996.

[11] Satava, R.M., “Virtual Reality and Telepresence for Military Medicine”, *Computer in Biomedicine*, 25(2), pp. 229-236, 1995.

[12] Arai, F., Tanumoto, M., Fukuda, T., Shimojima, K., Matuura, H.. and Negoro, M., “Distributed Virtual Environment for Intravascular Tele-Surgery Using Multimedia Telecommunication”, *IEEE Proceedings of VRAIS'96*, pp.79-85, 1996.

[13] Denning, A., “ActiveX Controls Inside Out”, *Microsoft Press*, 1997.

[14] Lorensen, W. E. and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *ACM Computer Graphics*, 21(4), pp.163-169, 1987.

[15] Tsai, M.D., Chang, W.C., Hsieh, M.S. and Wang, S.K, "Volume Manipulation Algorithms for Simulating Musculoskeletal Surgery", *Pacific Graphics'96*, pp.220-234, 1996.

[16]Koch, R.M., Gross, M.H., Carls, F.R., Buren, D.F.von, Fankhauser, G., Parish, Y.I.H., "Simulating Facial Surgery Using Finite Element Models", *ACM ComputerGraphics*, 30(5), pp.421-428, 1996.