An Orthopedic Virtual Reality Surgical Simulator

Ming-Dar Tsai[1], Shyan-Bin Jou[1], Ming-Shium Hsieh[2]

[1] Chung Yuan Christian University, Institute of Information and Computer Engineering
Chung Li, 32023 Taiwan, R.O.C.
[2] Taipei Medical College Hospital, Orthopaedics and Traumatology Department,
252, Wu Hsing Street, Taipei, Taiwan, R.O.C.
{tsai@ice.cycu/jou@earth.ice.cycu/shiemin@mail.tmc}.edu.tw

Abstract

This paper describes a highly interactive virtual reality orthopedic surgery simulator. The simulator can section, reposition and join volume-represented structures. By these functions, the simulator allows surgeons to use various surgical instruments to operate on virtual bones for simulating every procedure of complex orthopedic surgeries.

Key words: Virtual reality, Orthopedic surgery, Volume based visualization and simulation

1. Introduction

Orthopedic surgeries usually involve complex geometry and topology changes in bone morphology. Current training methods for interns and residents in teaching hospitals do not adequately raise spatial perception about the geometry and topology changes of bone morphology. Two reasons for the inability are trainees can only observe an operation before he participates in surgery and preoperative rehearsal usually involves 2D paper surgical simulations based on X-ray images. Orthopedic visiting doctors may also fail in real operations (e.g. 10%~20% for high tibia osteotomy [1,2] and 5%~15% for anterior fusion of the spine [3,4]) because there exists geometric and topology failures in bone morphology. These failures include false sections on bones, poor contact surfaces, inappropriate size and shape of bone graft and improper reduction position. The reason is considered the visiting doctors can only use 2D paper simulations to rehearse and confirm surgical plans.

The application of virtual reality (VR) to surgical training gives a more realistic human machine interaction than traditional 2 dimensional simulations and has already become a useful surgical planning and training tool. Several VR surgical simulators have been developed to provide detailed information regarding simulated tissues, tools and actions of surgeons [5].

VR simulation systems provide virtual environment by rendering a surface model that may be reconstructed from video data (for simulating endoscopic or laparoscopic surgery [6]), X-rays (for leg surgery [7]), or synthetic surfaces (for ophthalmic surgery [8]). However, the surface models are difficult to be employed to compute topology changes because of no interior information. Contrast to surface models, a volume (stack of 2-dimen-sional grayscale images) model represents a body as regularly partitioned cuboids (voxels) is suitable to reveal relations between tissues (with resolution limits but no projection errors [9, 10]) and simulate surgeries with topology changes.

Many excellent algorithms have been developed for visualizing a volume. For example, tissue surfaces can be well approximated by hundreds of thousands of triangulated isosurfaces [11]. These isosurfaces can be quickly rendered by current PC platforms. Some orthopedic simulators have employed the isosurfaces of anatomic structures to generate a virtual environment for training arthroscopy [12] and fixing [13]. However, manipulating the isosurfaces can not simulate orthopedic surgeries involving topology changes.

Surgical simulation algorithms usually manipulate voxels directly to simulate surgeries especially the ones with topology changes on structures. For example, most commercial imaging systems use a simple method of manipulating voxels, a cut-away operation to remove the voxels of one side of a cutting plane for removing obscurations. However, many cut away operations must be used for simulating a procedure (even a simple section) of orthopedic surgeries. Two approaches of manipulating voxel-represented objects have been discussed. One uses 2D pointer array to record the result of a series of cut away operations [14]. Then, adding the translation to the voxels in the lists can simulate the visual effects of translating a structure. Another approach extends the contents of each voxel, for example, 6 links to represent relations between a voxel and its six face neighbors. Adding or deleting links are easily implemented for local manipulations such as cutting or joining two objects [15]. However, link additions and deletions are time and memory consuming. Moreover, global manipulations such as repositioning objects or joining separate objects are difficult although they are also necessary in orthopedic surgeries.

This paper describes a VR orthopedic surgery simulation system manipulating volume data. To reposition a voxel-represented object, voxels are distinguished in a sub-tissue level. A structure code is assigned to achieve this purpose. By searching the voxels with the same structure code, all voxels of a structure can be traversed to manipulate (remove, reposition or assign as another structure to join). We have presented several algorithms that manipulate the structural voxels in 3D ways to simulate surgical procedures including cutting, identifying, removing and repositioning a structure, joining two structures into one, and testing collision during moving a structure. Combining these procedures, our system can provide surgical functions that operate a 3D image (virtual patient) as actual procedures on a real patient and ensure the accuracy of anatomic morphology in interactive responses. Through the 3D visual input and output environment, spatial perception of every procedure and its result give more effective simulations.

2. System Overview

The system was first reported in 1996 [16], and has since been modified and improved. The software is implemented in C++ (Visual C++ ver 5.0) under Microsoft Windows of a PC platform, and uses the OpenGL libraries to render isosurfaces without special graphics hardware. The PC must be equipped with a shutter glass and a tracker.

Figure 1 shows the system architecture. A user wears a shuttle eyeglass to observe stereographic images and uses a surgical instrument attached with a 6 dimensional degree tracker to simulate surgical procedures. The system includes an interface module, volume conversion module, isosurface reconstruction module, rendering module, and simulation module.

2.1 Interface module

The interface module provides virtual instruments and selectors including menus and data slide-bar. Using the menus, the user can choose a volume to simulate, determine a simulation function to operate, and input bone grafts and prostheses that have been designed by an AutoCAD system and change parameters of the shading model about light and material properties. Through the slide-bars, the user can easily change (slide) perspective conditions including viewing positions and angles, disparities of stereographic images to choose suitable ones.

The tracker is attached to one end of a surgical instrument to simulate a virtual instrument. Based on the position and attitude of the tracker and the shape data obtained from the instrument, the system can compute spatial data for the virtual instrument. Using the spatial data, the system can render the instrument to obtain its 3D image and compute the intersections between the instrument and the volume for simulating surgeries. The system currently provides the following virtual instruments: bone saw and osteotome for sectioning bone, virtual plate and staple for fixation, virtual dissector and currector for removing tumors, and virtual hand for moving bones, bone grafts and prostheses. The tracker is also used as a positioning instrument that partitions a volume into several subvolumes for the convenience in rendering tissue surfaces.

2.2 Data conversion module

For manipulating voxel-represented objects, every voxel is assigned three 6-bit distance-levels to simulate tissue surface changes, six 1-bit face codes indicating whether the voxel faces are on the boundary and one byte indicating a tissue type and structure number. A total of 4 bytes of memory are used for each voxel. Bone grafts and prostheses are designed by the AutoCAD system first, then converted to voxel-represented objects.

2.3 Isosurface reconstruction module and rendering modules

In contrast to thresholding techniques that determine a sample point on a tissue surface (isosurface) by one over-threshold voxel and one under-threshold voxel, one distance-measured voxel can determine a sample point [17]. Therefore, the three distance-levels are interpreted as three sample points on the three main axes respectively. Our system then use the marching cube algorithm that employ the sample points on the main axes to reconstruct triangulated isosurfaces [11].

2.4 Simulation module

The "section" function first interprets the positions and attitudes of a tracker as swept sectioning surfaces, computes distance-levels for sectioned boundary voxels, and assigns a structure code to the voxels. The "recognition" function identifies a separate structure, it uses an efficient 3D seed and flood algorithm to assign the voxels the same structure code inside a closed boundary composed of voxels with the same structure code (sectioned boundaries) and voxels of different tissues (natural boundaries). Unlike straightforward seed and flood algorithms that put six neighbors into a stack for recursion, voxels along some axis are directly computed and not stacked in the algorithm [18]. Therefore, voxels for recursion are considerably reduced.

The "removal" function assigns all voxels of a structure to be air voxels that can also be implemented by the 3D seed and flood algorithm. The "fusion" function re-recognizes one anatomic structure (separate bones, prostheses or bone grafts) from another and joins them together. The structures may contact each other and no new structure voxels are generated. New structural voxels may be generated to help in the fusing process. In this situation, this function generates closed boundary voxels between two user-specified curves on fusing structures. The system then recognizes the voxels inside the new boundary voxels with the old structures as one structure.

The "collision test" function detects collision among bones, prostheses, vessels and nerves. He proposed an efficient collision detection method that maps all objects into a map of regular cells, then detects collisions if objects occupy other object's spaces [19]. This grid intersection method was not adopted to detect collisions in our system because other functions are implemented during the collision test. One such function determines the distance between structures when a structure is placed onto another structure in a "fusing" simulation. The other function assigns a structure code to traversed soft-tissue voxels in a "healing" simulation. We adopted an efficient ray traversal algorithm to detect collisions (whether bone or nerve voxels exist on the path of a moving anatomic structure or surgical instrument). This algorithm is the most efficient because it has the fewest additions and comparisons [20].

The "reposition" function translates all voxels of a structure to another position by first implementing a "collision test" to detect collisions, then pushing the structure into a series of stacks and clearing the structure by the seed and flood algorithm before popping the structure to the new position. The three components of the translating vector are not limited integers. This means the system allows an unaligned translation that usually occurs when a structure is moved along the slice direction.

3. Results

In the following, we demonstrate two simulation examples operated by a visiting doctor. The CPU times were obtained under implementing on a PC with a Pentium-III 800 MHz CPU and 256 Mbytes of main memory.

3.1 Arthroplastic simulation example

In the simulation of arthroplasty operations, the user sections (using the "section" function) bones until one or more anatomic structures are separated (and thus recognized by the "recognition" function) from the skeleton. The user may remove (using the "removal" function) the structures to correct the skeletal morphology, to accommodate the prosthesis or in the case where the structures are abnormal bones. A prosthesis is used to replace a removed joint. The surgeon may reposition the structures (using the "reposition" function) to correct the skeletal morphology, and then fix the structures and fuse (using the "fusing" function) them into the skeleton.

Figure 2 shows the image rendering results of an example knee arthroplasty operation that was performed to replace a destroyed joint and correct a malposition of the tibia. The volume was constructed with 24 CT slices at a 256×256 resolution. However, we enlarged the volume as 35×256×256 resolution for manipulating a user-input prosthesis. The computation time was 2.2 seconds to reconstruct the bone isosurfaces for the whole volume and 0.29 seconds to obtain a 3D image with this system. After the prosthesis was input into the volume, the isosurface reconstruction time becomes 3.2 seconds for the bone and prosthesis isosurfaces and the rendition time becomes 0.42 seconds.

Figure 2(a) shows the proximal tibia being sectioned by the saw. The interface slidebars for determining the parameters of the various perspectives and menus for determining a simulation function are also shown in the left top and bottom respectively. Figure 2(b) shows the results after two flat sections on the femur and tibia respectively, followed by recognition and removal of a near flat bone fragment of the femur and a wedge-shaped fragment of the tibia. A hand (a virtual instrument) began to reposition the tibia. Figure 2(c) shows the tibia was repositioned to correct the mal-position. A vertical bone fragment was sectioned away so that the femur can accommodate the posterior of the prosthetic femur. The virtual hand was removing the bone fragment. Figure 2(d) shows a vertical bone fragment and an oblique one were sectioned away such that the femur can accommodate the anterior of the U-shaped prosthetic femur. An oblique section on the posterior of the femur for accommodating the U-shaped prosthetic femur and an oblique section on the patella for accommodating the prosthetic patella were then sectioned away.

Figure 2(e) shows the prosthesis has been recognized (by recognition functions) as three separate structures: a curved femur part, disk-like tibia part and dome-like patella part. The tibia part has been repositioned for insertion on the tibia by the virtual hand. This figure also shows the dome of the prosthetic patella can well slide inside the groove of the prosthetic femur and the prosthetic tibia also well matched to the tibia plateau. The three prosthetic parts are good choices for working the knee functions. Figure 2(f) shows the U-shaped prosthetic femur has been repositioned for insertion on the femur. The prosthetic patella has also been repositioned for the insertion on the patella. However, we can not observe it well because it is almost hidden by the patella. The prosthetic femur well accommodated to the femur, therefore the previous sections on the femur were appropriate. The U-shaped curve of the prosthetic femur can also slide well inside the grooves of the prosthetic tibia. Therefore, the prosthetic femur and tibia are considered well positioned.

The simulation example provides an anatomical demonstration that the knee arthroplasty can correct the mal-position of the tibia, accommodate the tibia and femur to fit the prosthesis and insert the prosthesis into the correct position. The complex changes in bone morphology involved in this surgery were well simulated by our system. The results of every procedure can be thoroughly demonstrated with a high-quality 3D image. Table 1 shows the computer response times for the simulations involved in the knee arthroplasty. A complete simulation is defined as including completion of the specified function, reconstruction of the isosurfaces and rendering of the corresponding image. Because most the simulations responded in 2 seconds, we considered our system could achieve the requirement of interactive responses.

3.2 Open ostetomic simulation example

Open osteotomy is used to open bone in order to remove tumors inside the bone. Upon simulation of this technique using our system, the user sections a bone until a window structure separates. He then repositions the structure away to indicate opening the bone by using the recognition function and then the reposition function. Then, he dissects the tumor (using the section function) and removes it. The user may simulate implantation of a bone graft by inputting a bone graft and repositioning it to the tumor position. The user then finally repositions the separate window structure to the original position and fuses the window structure with the original bone together to simulate closure of the bone.

Figure 3 shows the rendering results of a knee open osteotomy for removing a tumor inside the proximal tibia. The volume was constructed with 28 CT slices at a resolution of 256×256. Figure 3(a) shows a 3D image of a knee where the proximal tibia was being sectioned by a virtual saw. Figure 3(b) shows a window-shaped bone fragment that has been sectioned, recognized and repositioned away using the virtual hand. The area of the tumor is marked with an orange color. A dissector (indicated by a red color) is available to dissect the tumor. Figure 3(c) shows the tumor being dissected by the dissector. Figure 3(d) shows that the tumor has been removed and a graft bone (lower left corner) has been prepared (already recognized) to fill the space of the resected tumor. Figure 3(e) shows that the graft bone has been implanted and that the window-shaped bone fragment was being repositioned again to its original location. Figure 3(f) shows the results after fusing the bone fragment with the knee. The results suggest that the position and size of the window fragment is a reasonable choice for opening the knee and allowing the tumor to be completely removed. The graft bone is suitable to fill up the tumor space.

4. Conclusion and Future Work

Computed tomography (CT) or magnetic resonance imaging (MRI) scanning has become a standard procedure to reveal interior anatomies. Visualizing a volume constituted by transversal slices can ease observation of anatomies to improve diagnoses. Beyond the volume visualization, manipulating volume data to simulate deformation or topology changes of tissues during a surgery can verify surgical plans, rehearse procedures and predict prognoses.

Our simulation methods can manipulate voxel-represented bone structures to model interactions between the bones such as: cutting, fusing, repositioning, recognition, and collision testing of a moving bone. By these functions, our system can simulate complex geometry and topology changes of bone morphology for every orthopedic procedure. These capabilities are necessary to provide helpful spatial information for most orthopedic surgeries. Therefore, the simulator is useful in the preparation for many kinds of difficult surgical procedures that are often performed in the orthopedics department without putting patients at risk.

The future works can focus on improving some drawbacks of the prototype system. Improvements in the user interface can ease users to operate the system. For example, we hope to assign distinct colors to different structures for high-lighting some structure. As the arthroplasty example shows, the bones and the prosthesis are considered as the same tissue because they can be fused together. The more important prosthesis is better to be high-lighted for easy observation. The improvement in reducing response time of the system would help improving realism and ease to use. Because the system uses a volume to simulate every surgical procedure, rendering isosurfaces reconstructed from a volume to obtain a 3D image is usually computationally demanding. Use of decimation techniques to reduce triangles of isosurfaces or other volume visualization techniques such as accelerated volume rendering techniques can be tried to save the rendition time.

Acknowledgment

The authors would like to thank the National Science Council for financial support of this research under Contract No. NSC-86-2213-E-033-036, NSC-87-2213-E-033-005, NSC-89-2320-B-038-019-M08 and NSC-89-2213-E-033-070. We would also like to thank all of our colleagues at the computer research lab, the visiting doctors, residents and interns at our orthopedic department for participation in the study. We also like to thank the Radiology Department of Taipei Medical College Hospital for providing the CT and MRI data and the United Orthopedic Corporation of Taiwan for providing the prosthesis data.

References

1 J.A. Finkelstein, A.E. Gross, A. Davis, Varus osteotomy of the distal part of the femur: A survivorship analysis, J. Bone Joint Surg. Am. 78A (1996) 1348-1352.

2 H. James, Beaty, Orthopaedic Knowledge Update. In: San. Howard, T.L. Randall (Eds.), Spine. Rosemont, American Academy of Orthopaedic Surgeons, 1999, pp.561-754.

3. J.J. Regan, H. Yuan , P.C. Mc Afee, Laparoscopic fusion of the lumbar spine: minimally invasive spine surgery, a prospective multicenter study evaluating open and laparoscopic lumbar fusion, Spine 24 (1999) 402-411.

4. M. Muschik, H. Zippel, C. Perka, Surgical management of severe spondylolisthesis in children and adolescents. Anterior fusion in situ versus anterior spondylodesis with posterior transpedicular instrumentation and reduction, Spine 22(17) (1997) 2036-2042.

5. R.M. Satava, Virtual reality, the current status of the future, in: S. Weghorst, H. Siegurg, K. Morgan (Eds.) Health care in the information Age, IOS Press, Amsterdam, 1996, pp. 542-545.

6. R.M. Satava, Virtual reality surgical simulator: The first steps, Surgical Endoscopy 7(3) (1993) 203-205.

7. L. Caponetti, A.M. Fanell, Computer-Aided Simulation for Bone Surgery, IEEE Computer Graphics and Applications 13(6) (1993) 87-91.

8. I.W. Hunter, L.A. Jones, M.A. Sagar, S.R. Lafontaine, P.J. Hunter, Ophthalmic Microsurgical Robot and Associated Virtual Environment, Computer in Biology and Medicine 25(2) (1995) 173-182.

9. S. Seipel, I.V. Wagner, S. Koch, W. Schneide, Oral implant treatment planning in a virtual reality environment, Computer Methods and Programs in Biomedicine 57(2) (1999) 95-103.

10. S. Baumrind, F. Moffit, S. Curry, The Geometry of Three-Dimensional Measurement from Paired Coplanar X-Ray Images, Am. J. Orthod., 84 (1983) 313-326.

11. W.E. Lorensen, H.E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, ACM Computer Graphics, 1987, pp.163-169.

12. R. Ziegler, G. Fischer, W. Muller, M. Gobel, Virtual Reality Arthroscopy Training Simulator, Computer in Biology and Medicine 25(2) (1995) 193-203.

13. A. Sourin, O. Sourin, H.T. Sen, Virtual Orthopedic Surgery Training, IEEE Computer Graphics and Applications, 20(3) (2000) 6-9.

14. J.K. Udupa, D. Odhner, Fast Visualization, Manipulation and Analysis of Binary Volumetric Objects, IEEE Computer Graphics and Applications, 11(5) (1991) 53-63.

15. S. F. Frisken-Gibson, Using Linked Volumes to Model Object Collision, Deformation, Cutting, Carving, and Joining, IEEE Transaction on Visualization and Computer Graphics 5(4) (1999) 333-348.

16. M.D. Tsai, M.S. Hsieh, W.C. Chang, S.K. Wang,, Volume Manipulation Algorithms for Simulating Musculoskeletal Surgery, Proceedings of the Fourth Pacific Conference on Graphics and Applications, 1996, pp. 220-234.

17. G. Sealy, K. Novins, Effective Volume Sampling of Solid Models using Distance Measures, roceedings of IEEE Computer Graphics International, 1999, pp.26-32.

18. L. Feng, S.H. Soon, An Effective 3D Seed Fill Algorithm, Computer & Graphics 22(5) (1998) 641-644.

19. T. He, A. Kaufman, Collision Detection for Volumetric Objects, IEEE Computer Society Press, Proceedings of IEEE Visualization, 1997, pp.159-166.

20. D. Cohen, A. Kaufman, 3D Line Voxelization and Connectivity Control, IEEE Computer Graphics and Applications 17(6) (1997) 80-87.

Table 1 Total simulation time for knee arthoplasty simulations

	1	2	3	4	5	6	7	8	9	10	11	12
Time spent(sec)	0.95	1.24	1.98	0.96	1.95	0.93	1.88	1.97	1.75	2.05	3.20	0.93
Voxels involved	8879	11845	37365	5252	4990	4880	3985	3754	4750	19605	39630	1250

1. sectioning away (sectioning, recognizing and removing) the high-tibia; 2. sectioning away the low-femur; 3. recognizing and repositioning the tibia; 4. a vertical section on the anterior part of the femur; 5. recognizing and removing the vertical section; 6. an oblique section on the anterior part of the femur; 7. recognizing and removing the oblique part; 8. A vertical section on the posterior part of the femur then moved away; 9. an oblique section on the posterior part, then moved away; 10. repositioning the prosthetic tibia; 11. repositioning the prosthetic femur; 12. repositioning the prosthetic patella.

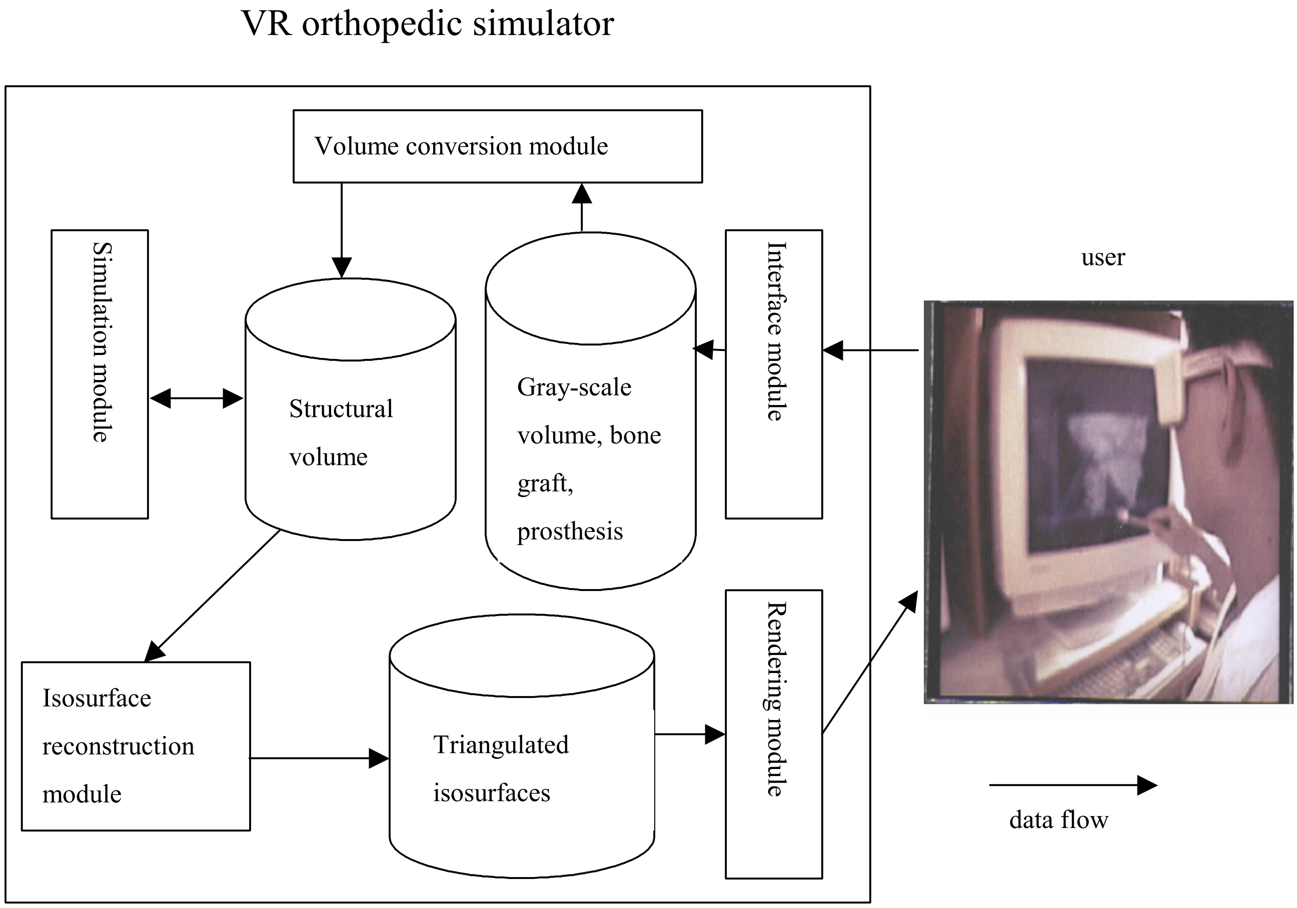
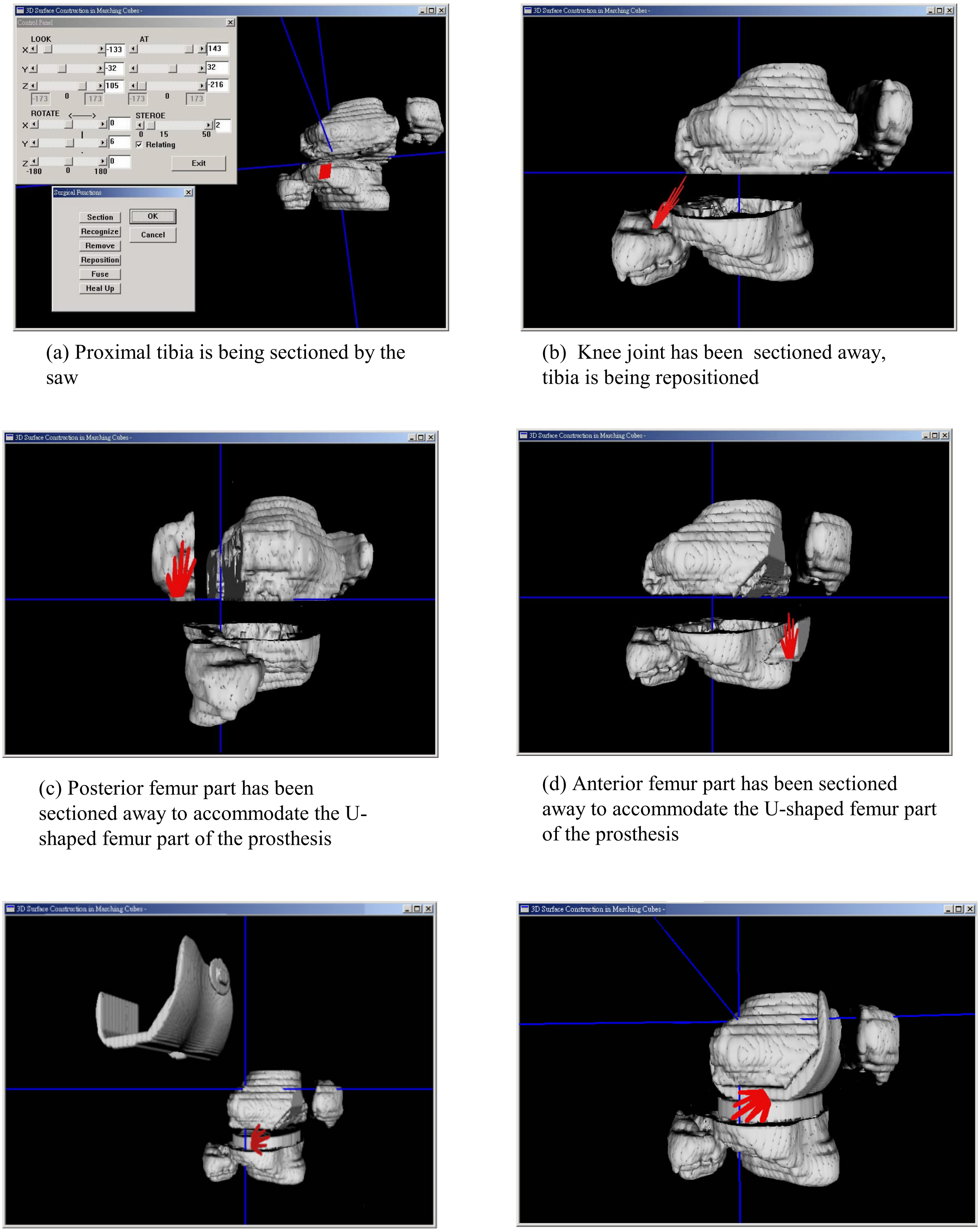


Figure 1 System architecture



LOOK X -133, Y -32, Z 105 (-173 0 173)

AT X 143, Y 32, Z -216 (-173 0 173)

ROTATE X 0, Y 6, Z 0 (-180 0 180)

STEROE 2 (0 15 50)

☑ Relating

(a) Proximal tibia is being sectioned by the saw

(b) Knee joint has been sectioned away, tibia is being repositioned

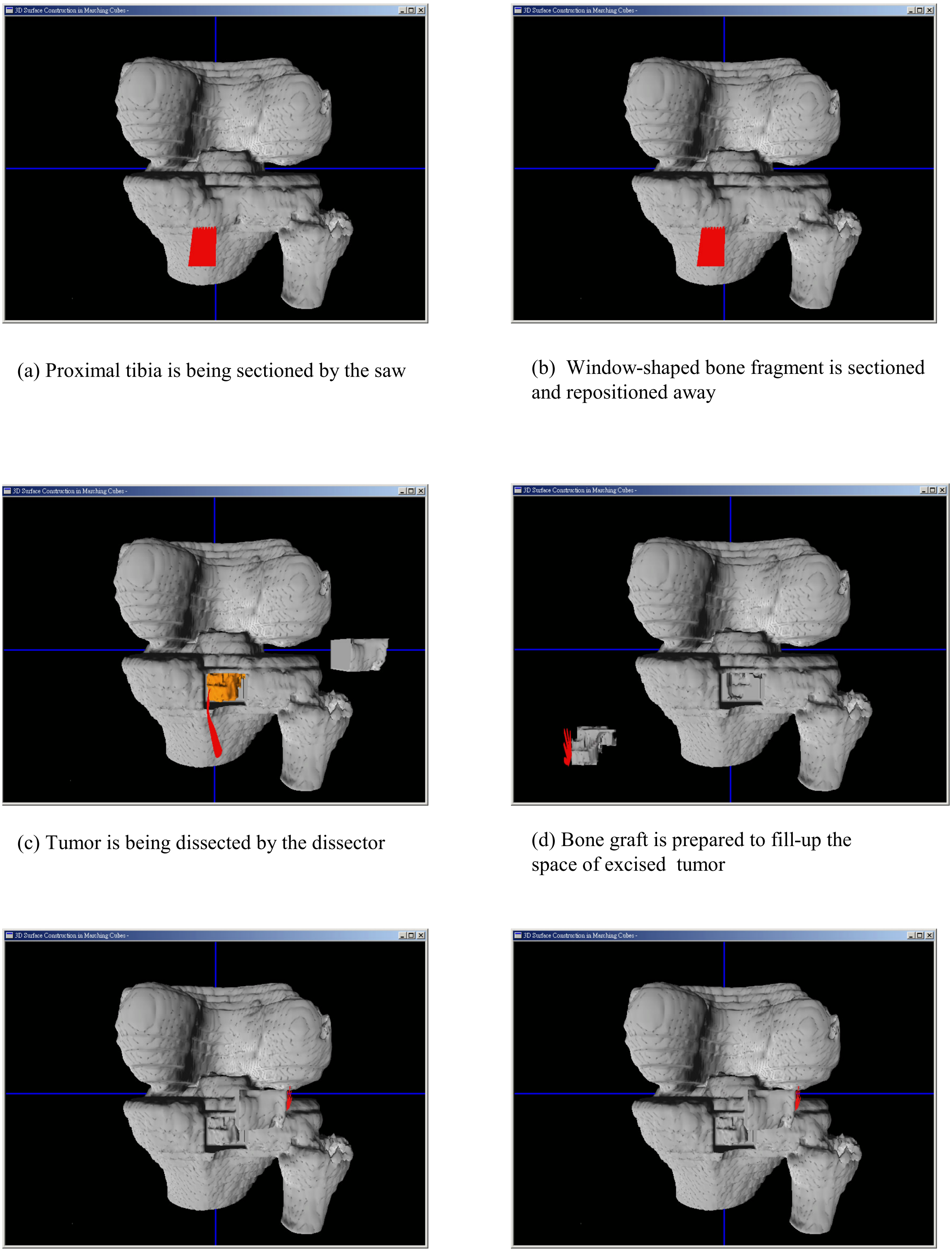
(c) Posterior femur part has been sectioned away to accommodate the U-shaped femur part of the prosthesis

(d) Anterior femur part has been sectioned away to accommodate the U-shaped femur part of the prosthesis

(e) Tibial part of the prosthesis has been inserted into to the knee joint

(f) Femur part of the prosthesis has been inserted into the knee joint

Figure 2 Arthoplasty for replacing the knee joint and correcting mal-position of the knee



(a) Proximal tibia is being sectioned by the saw

(b) Window-shaped bone fragment is sectioned and repositioned away

(c) Tumor is being dissected by the dissector

(d) Bone graft is prepared to fill-up the space of excised tumor

(e) Bone graft has been implanted, and the window shaped piece of bone is being repositioned to the original position

(f) Window shaped bone is repositioned and fused with the tibia

Figure 3 Open osteotomy for removing a tumor in the tibia