UNIFYING DATA STRUCTURES FOR VIRTUAL REALITY APPLICATIONS

Ícaro Lins Leitão da Cunha [1], Ronei Marcos de Moraes [2], Liliane dos Santos Machado [3]

Abstract *— In the past few years, Virtual Reality Systems (VRS) have been used as an approach for the development of effective training systems. The CyberMed is a free software system that offers libraries for the development of simulation applications for medical teaching and training based on VRS. The system includes functionalities as stereoscopic visualization, haptic interaction, objects deformation and user evaluation. To share the use of 3D objects in these functionalities it's necessary to create a storage structure that favors the operations made by them. By this means, the storage of the information by data structures (DS) based on corner-tables shows quite adequate due to the fact that it stores references to the neighboring points of a triangle that forms the object. The direct access to a triangle's neighbors is necessary to guarantee the real-time feature on deformation and illumination routines.*

Index Terms — Corner-table, Data Structure, Unifying Data, Visualization.

INTRODUCTION

Surgery simulation is an essential tool to acquire experience and ability because can decrease the time of recovery of patients and the number of mortalities in surgeries. For its existence, a realistic simulation system depends on the quality of its images, the non-existence of delays in the simulation and the interaction methods offered by the system [1]. Having a real-time interaction and visualization means that any action made by the user generates an instantaneous response by the system, modifying its components and giving back reactions to the user. By these means, the complexity of the objects and the events triggered by the interactions should not affect the system's performance.

Virtual Reality (VR) is a recent research area that unites knowledge of many areas such as electronics, computer science, robotics, physics etc. The objective of VR is to offer real-time systems that integrate aspects of immersion and interactivity to simulate realistic environments [2]. For this, specific equipments are used to stimulate the senses of sight, hearing, touch and smell [5][6].

CyberMed is a VR system to support medical teaching and training by user interactions [3]. It allows the exploring of the human body and the realistic simulation of medical proceedings in an immersive virtual environment. CyberMed system is developed under the VirtWall platform [4], a virtual wall based on public domain tools and low cost equipments. This feature is important to make the use of CyberMed economically possible in emergent countries, such as Brazil. The main features of CyberMed are: three-dimensional visualization; usage of realistic models; spatial interaction; interactive deformation; visual sharing; and supervision and evaluation [3].

This paper presents the development of the DS shared by the routines of the CyberMed. The main motivation of this is to generate a DS capable of storing data from graphical objects and allowing their fast access by the different routines of the system to guarantee the system's real-time feature.

CORNER-TABLE DATA STRUCTURE

A corner-table is a concise data structure for representing triangular meshes. In it, the concept of corners is used to represent the association of a triangle to one of it's vertexes, in other words, associate a triangle to a opposite border to a corner [7][8].

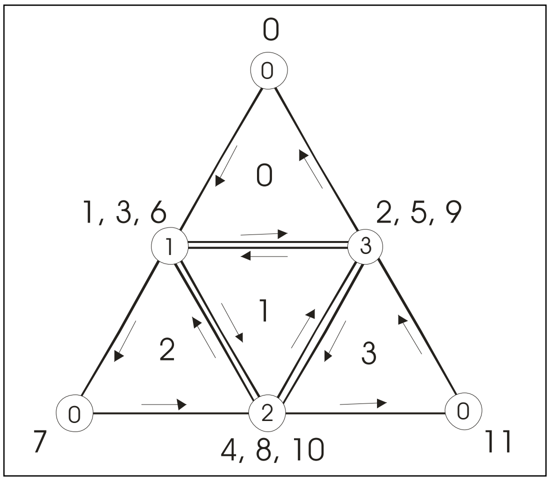

FIGURE. 1
VERTEXES AND FACES FLATTENED TETRAHEDRON.

[1] Ícaro Lins Leitão da Cunha, scientific initiation student financed by CNPq, Department of Informatics, Universidade Federal da Paraíba – UFPB, Cidade Universitária s/n, João Pessoa, PB - Brazil, ivellius@yahoo.com.br

[2] Ronei Marcos de Moraes, Department of Statistics, Universidade Federal da Paraíba – UFPB Cidade Universitária s/n, João Pessoa, PB - Brazil, ronei@de.ufpb.br

[3] Liliane dos Santos Machado, Department of Informatics, Universidade Federal da Paraíba – UFPB, Cidade Universitária s/n, João Pessoa, PB - Brazil, liliane@di.ufpb.br

In the DS, non-negative numbers indexes corners, vertexes and triangles. Three consecutive corners define each triangle, and their order defines its construction orientation. For example, in figure 1 the corners 0, 1 and 2 represent the first triangle (triangle 1), the corners 3, 4 and 5 the second triangle (triangle 2) and so on. As a consequence to this, a corner *c* is associated to it's triangle of index *t* by the formula $t = c \% 3$, where % is an operation that gives the rest of a division, in this case the rest of the division of *c* by *3*. The corner-table represents the geometry of a surface by associating each corner *c* to its geometric vertex index.

Assuming the counter clockwise construction orientation of the triangles, for each corner *c*, the next(c) and before(c) are corners that can be found by the following equations: $prox(c) = (c+1) \bmod 3$ and $before(c) = (c+2) \bmod 3$.

By associating to each corner *c* its opposite corner *O[c]* that shares the same opposite border, the adjacency of borders of neighboring triangles is represented. Table I shows the list of data of the object in Figure 1 generated by corner-table.

TABLE I

LIST OF DATA OF TETRAHEDRON.

Corner	Vertex	Triangle	Opposite
0	0	0	2
1	1	0	2
2	3	0	2
3	1	1	0
4	2	1	0
5	3	1	0
6	1	2	3
7	0	2	3
8	2	2	3
9	3	3	1
10	2	3	1
11	0	3	1

UNIFICATION AND DIRECT ACCESS OF DATA IN THE CYBERMED

CyberMed is a VR based system that supports applications for medical teaching and training [3]. Interactive exploration of the human anatomy and realistic simulation of medical procedures in an immersive virtual environment are some of its applications.

The main features of CyberMed are: three-dimensional visualization of graphical objects; usage of realistic models generated from 3D reconstruction of medical images; spatial interaction with touch detection, using tactile devices; interactive deformation of objects; visual sharing in case of using the system on a class room, by this the visualization can be seen by more than one users; and supervision and evaluation of the user's actions.

Triangular meshes form the graphical objects used in CyberMed using triangular faces to define a polyhedral surface proves to be more convenient than using arbitrary polygons. This is true because of the simplicity of triangles, since they offer planar advantages; they are always polygonal plane curves and have linear coordinates (barycentric coordinates) naturally associated to them.

Each object used in CyberMed has associated to them a series of physical properties; these properties are used by the routines in CyberMed. To make them organized and accessible to these routines they have to be stored in a DS, this can make their accesses more fast.

In an interactive simulation on the system, a user can interact with the environment by changing the position of a pointing object (using either mouse or haptic device). To this interaction there is a routine associated to it that verifies if there is a collision of this object with the other objects of the 3D environment. For this collision test it is necessary to identify the position of the pointing object in relation to the objects on the scene. In case a collision occurs, deformation, touch, visualization and evaluation routines are activated to make the necessary calculations to give a feedback to the user interaction. For this happen, much information regarding the 3D objects in the scene is necessary, this information can be: references to a triangles neighbor and elastic coefficients for deformation calculations and return of tactile force, vertexes and triangles (including neighboring triangles) normal vectors for illumination calculation.

Figure 2 presents the association of the DS with the view, deformation, touch and evaluation routines of CyberMed.

Erro!

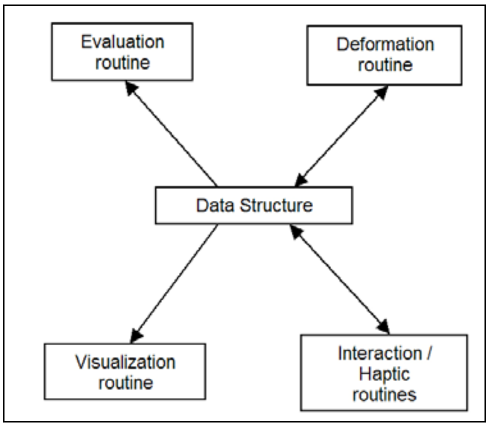

FIGURE. 2

INTEGRATION OF THE SYSTEM'S FUNCTIONS WITH THE DS.

Any delay made in the access to the necessary data of CyberMed's routines (like long searches in the DS and constant references between different DSs) can be fatal for the system's real-time feature.

This way, not only points and polygons of a 3D object are necessary, but the system also needs values, such as triangles' neighbors, color of triangles and their normal vectors, amongst others, for the execution of its routines. These values can be calculated or obtained only once and are stored in a DS.

So it is important that the DS gives a direct access to the original data of the 3d objects and also the calculated data related to these objects.

DATA STRUCTURE GENERATION

The use of a 3D object by the CyberMed is done by reading from the source file of a model that contains the object description. The object data is stored in a corner-table based DS. The original data is used to calculate the other data necessary for the system processing such as color of the objects, normal vectors of vertexes and polygons, elasticity constants of the vertexes' bonds (edges of the triangle), the mass of the points, and other constants**.**

The CyberMed DS is based on the corner-table DS with a feel adjustments: data like coordinates and neighboring triangles are accessed more directly in it's own triangle (different to a list of sequenced points), with out the need of numeric indexes; the storage of more data is applied; and the DS is referencing neighboring triangles, not opposite points, by this the whole data of the neighboring triangle can be accessed without the need of a linear search for it. These changes are necessary to make faster accesses to information and by doing so improving the performance of the routines in CyberMed.

The source file that is being used is in the VRML 2.0 format. It presents the following geometric properties of the model: color, points and polygons. Other data like normal vectors and neighboring triangles are calculated when the DS is being constructed. In all, the file contains two main parts: a list of points and a list of polygons.

On the list of the points the x, y and z coordinates are stored, the value of these coordinates are in floating point. For the tetrahedron in Figure 1, this part is presented as:

```
point[
   x0 y0 z0,
   x1 y1 z1,
   x2 y2 z2,
   x3 y3 z3
]
```

The list of polygons stores the reference to the points in the point's list and the construction orientation of each polygon. The point's references follow the order of each point in the point list and its value is in integer. The construction orientation of the polygon can have two values, either "1", if the orientation is clockwise, or "-1", if the orientation is counter clockwise. For figure 1 this part is presented as:

```
coordIndex[
   0 1 3 -1,
   1 2 3 -1,
   0 2 1 -1,
   0 3 2 -1
]
```

The first time the system is executed, the model file is read and stored. After that this data is used to calculate the neighboring triangles one in relation to each vertex of the triangle. So, in total, it calculates three neighboring triangles per triangle.

The DS stores a dynamic list of triangles and for each triangle it stores: the coordinates of each vertex, it's construction orientation, three pointers that references the triangle's neighbors, a pointer that references the next triangle on the list and other physical data of the object that will be used by the different routines. Figure 3 shows an example of the DS for the figure 1. The reference to the neighbor is stored on the same order as the point of the triangle that's opposite to it. By this and the fact that it's on the same structure as the rest of the object's data it is assured that there will be faster searches made by the routines within the structure.

The calculations for the determination of the neighboring triangles is done by a linear search in the polygon's list, for each triangle three different pairs of vertexes are selected and with them is made a search for another triangle that has one of the pairs, the triangle that is found is one of the neighboring triangles.

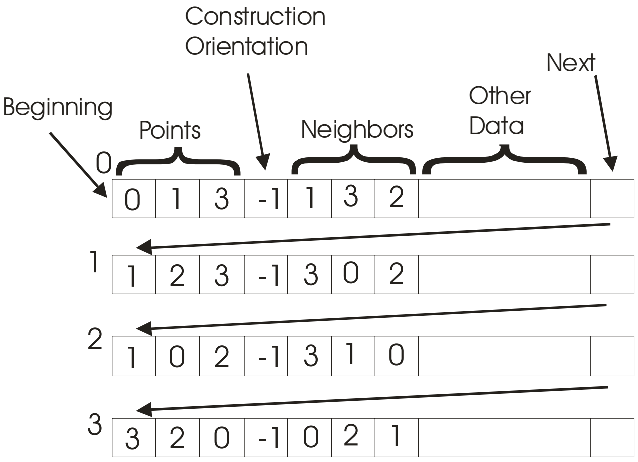

FIGURE. 3
DATA STRUCTURE OF THE OBJECT REPRESENTED IN FIGURE 1.

CONCLUSIONS

The generation of the DS was tested with three graphical objects: a simple cube with 8 points and 12 triangular faces; a medium sized sphere with 114 points and 224 triangular faces; a model that represents the marrow with 8036 points and 16072 triangular faces; and a model that represents de iliac bone with 12070 points and 24164 triangular faces.

Table II shows a graph that compares the time needed to calculate neighbors for each triangle of the four graphical objects used. The hardware used for tests was a PC AMD64 3.2Mhz with 1.5Gb of RAM running Fedora Core 3 32 bits.

TABLE II
TIMING TESTS

Model	Number of Vertexes	Number of Triangles	Time of Neighbor Calculation (in seconds)
Cube	8	12	0,00
Sphere	114	224	0,23
Bone Marrow	8036	16072	129,42
Iliac Bone	12070	24164	284,47

Figure 4 shows one of the models selected to test the data structure. Figure 5 shows a bone marrow model of the bone marrow presented inside a real iliac bone. The integration of both models can be observed in Figure 6. Because for each triangle there will be more data associated to it than the ones on the source file, this model was chosen because of its size to test the storage capacity of the DS. It's important to verify this capacity because the CyberMed will allows simultaneous visualization of more complex graphical objects. At the same time, the system will support multi-layer compositions (Figure 6), what means that will be on DS for each layer.

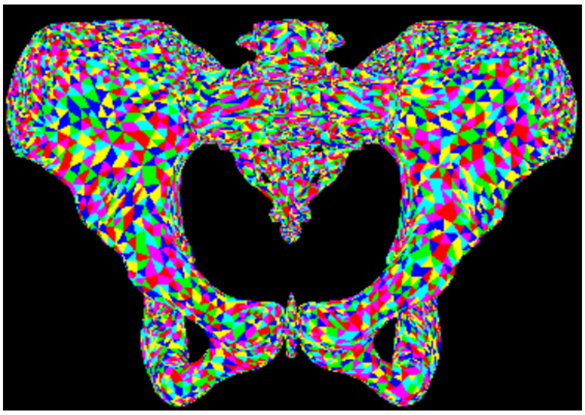

FIGURE. 4
TEST OF THE DATA STRUCTURE USING AN ILIAC BONE MODEL. THE 3D OBJECT HAS 12070 POINTS AND 24164 TRIANGULAR FACES. THE OBJECT IS COLORED TO DISTINGUISH THE FACES OF THE SURFACE.

The calculations for finding the neighboring triangles and the normal vectors for the polygons and vertexes are costly. Since these calculations will only be done on the loading phase of the model(s), this fact will only affect the loading phase of the VR system. Also these calculations will only be done when the model is loaded for the first time in the system, because after that the data is saved on a file for later use.

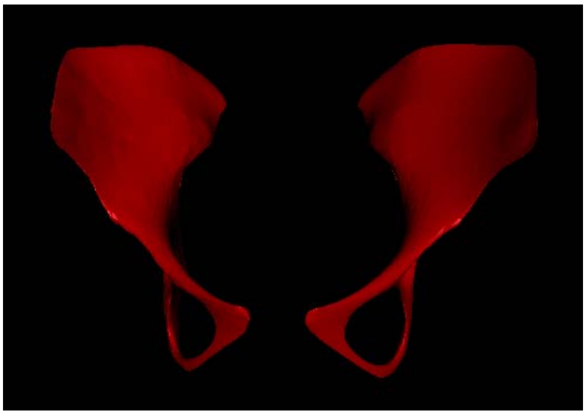

FIGURE. 5
THE BONE MARROW MODEL USED IN TESTS OF THE DATA STRUCTURE GENERATION.

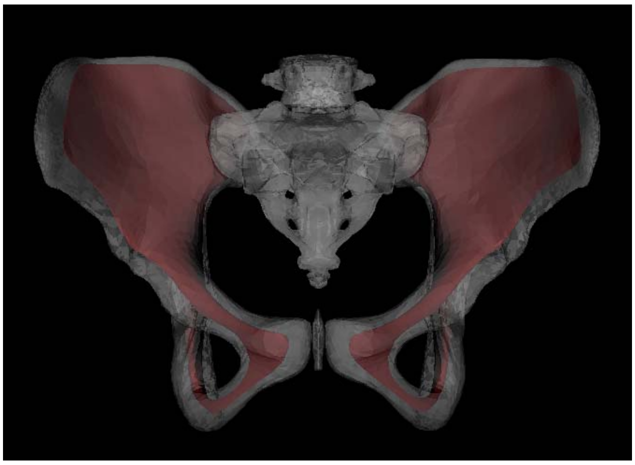

FIGURE. 6
BONE AND BONE MARROW MODELS VISUALIZED TOGHETER.

ACKNOWLEDGMENTS

This work was partially supported by CNPq (CT-INFO-CNPq 506480/2004-6) and FINEP (ref. 1898/04).

REFERENCES

[1] Basdogan, C.; De, S.; Kim, J.; Muniyandi, M.; Kim, H., et al., "Haptic Minimally Invasive Surgical Simulation and Training", *IEEE Computer Graphics and Applications*, Vol. 24, No. 2. March-April 2004.

[2] Machado, L.S.; Moraes, R.M. "Cenários 3D Interativos com Software Livre". *Revista de Informática Teórica e Aplicada.* Vol. 12, No. 2. 2005, pp. 91-112.

[3] Machado, L.S.; Cunha, I.; Campos, S.; Moraes, R., "CYBERMED: Realidade Virtual para Ensino Médico", *IFMBE Proceedings*. Vol. 5. 2004, pp. 573-576.

[4] Moraes, R.M.; Machado, L.S.; Souza, A.C.M., “VirtWall: A Concept of Low-Cost Virtual Wall for Immersion in Virtual Reality”, *Proc. Simp. on VR*. 2003, pp. 383-385.

[5] Netto, A.V.; Machado, L.S.; Oliveira, M.C.F., “Realidade Virtual - Fundamentos e Aplicações”. Visual Books, 2002.

[6] Rosemblum, L., Burdea, G. and Tachi, “VR Reborn”, *CG & Applications*, Vol. 18, No 6. 1998, pp. 21-23.

[7] Rossignac, J., Safanova, A., and Szymczak, A., “3D Compression Made Simple: Edgebreaker on a Córner-Table”. *Proceedings of the Shape Modeling International Conference*. May 2001.

[8] Vieira A.; Velho L.; Lopes H.; Tavares G. and Lewiner T. “Fast Stellar Mesh Simplification”, *Brazilian Symposium on Computer Graphics and Image Processing*. 2003, pp. 27-34.