DE GRUYTER OPEN

DOI: 10.2478/aucts-2014-0007

ACTA UIVERSITATIS CIBINIENSIS – TECHNICAL SERIES

Vol. LXIV No 1 2014

OPTIMIZATION OF COLLISION DETECTION IN SURGICAL SIMULATIONS

DAN CUSTURĂ-CRĂCIUN
UPB Bucharest Romania, No.5 Iuliu Gall str. Bucharest Romania, custura_dan@yahoo.com

DANIEL COCHIOR
„Titu Maiorescu" University, Department of Surgical Disciplines, Faculty of Medicine, Surgical Department Hospital CF 2, Bucharest, Romania, cochior_daniel@yahoo.com

CORNELIU NEAGU
Department of Manufacturing Technology, Politehnica University of Bucharest, Romania, neagu_corneliu@yahoo.com

Abstract: *Just like flight and spaceship simulators already represent a standard, we expect that soon enough, surgical simulators should become a standard in medical applications. A simulations quality is strongly related to the image quality as well as the degree of realism of the simulation. Increased quality requires increased resolution, increased representation speed but more important, a larger amount of mathematical equations. To make it possible, not only that we need more efficient computers, but especially more calculation process optimizations. A simulator executes one of the most complex sets of calculations each time it detects a contact between the virtual objects, therefore optimization of collision detection is fatal for the work-speed of a simulator and hence in its quality.*

Keywords: surgical virtual reality, collision detection optimization, representation quality.

1. Introduction

Traditionally, applying new surgical techniques involves certain interaction between the surgeon and his patient, more specifically his tissues (the barrier between the surgeon and his patient being represented at most by a surgical instrument) and gained handiness can be improved only by amplifying perceptions (by optical devices). The fusion between calculation technique, robotics, haptic devices, telecommunications and virtual reality (VR) makes possible a whole series of events such as: real-time virtual surgical intervention, allowing the surgeon to study various surgical procedures and to „replay" certain surgical-states or to complete his surgical training. Furthermore, VR makes „distance-surgery" accessible as well as providing robot-aid to the main surgeon during his surgical intervention. The first steps towards achieving this were already taken, meaning that surgical interventions were executed from outside the operating theaters, from another hospital or from another continent (Cochior, 2005, p. 167). Using virtual reality, by digitally transposing the surgical procedures, shall bring a revolutionary upgrade to surgery. It opens new possibilities of applying much more effective training methods; it allows experiment of new surgical techniques as well as it improves the surgeon's skills and abilities. During the past decade, we noticed a significant increase in the interest towards simulators development based on VR. In attempt to better understanding and increasing VR based simulators quality, research in subjects like virtual 3D geometry and computer graphics was intensely sustained and supported (El Saddik, 2011, p. 218).

2. Collisions; Algorithms

For a good representation of virtual reality, we refer to the role of the geometric model which is to stock geometric properties of bodies in the virtual environment (Zhou, 1995, p. 355-364). The geometric role model completed with the dynamic model will describe a body's or a group of bodies' behavior in movement tending to an equilibrium position inside a virtual environment (Miller, 2007; Masutani, 2004; James, 1999). Intervention coming from a human operator towards an object inside virtual space and usage

© 2014 "Lucian Blaga" University of Sibiu

of that object (instrument) in order to move, touch, deform or cut another object from the virtual space is based on collision detection, collision localization determination, the normal's and the reaction's local force determination (Ericson, 2005; Cavusoglu, 2004; Berkley, 2004)

In virtual reality, collision detection has a primordial role, being the base of virtual object-interaction representation process (Zachmann, 2000).

In terms of volume, in a surgical virtual reality simulator calculations must be performed by the following steps:

- Determination of Xmax for object 1
- determination Xmin for object 1
- determination Ymax for object 1
- determination Ymin for object 1
- determination Zmax for object 1
- determination Zmin for object 1
- ... determination X,Y,Z max and min for all the other objects
- Testing (6) on each pair of objects
- Choosing which pair of objects are in collision
- Contact surface/spot determination (GJK)
- Applied forces determination
- Distributed forces determination
- reaction forces determination
- deformation determination
- force feedback determination
- 3D environment representation
- 2D screen representation of visible parts in the environment

One of most used collision test is named AABBs (axes aligned bounding boxes) (Coumans, 2010). Unfortunatelly this test can't be used well in a case with complicated objects.

One of the most efficient algorithms is the **Gilbert-Johnson-Keerthi** (GJK) algorithm (Zhou, 1995; Gilbert, 1988). This algorithm calculates the distance between a random convex object and an exterior point iteratively. Moreover, it can specify which of the object's apex, edge or face is the closest to a certain point. Even though the algorithm is highly efficient, there is a probability that the solution to be found after 3 to 8 steps, each step requiring a series of calculations to be made.

Extrapolating, to tell if two convex objects are colliding, we should apply GJK algorithm to each object and moreover, to every point of the other objects. During the simulation of a laparoscopic surgical intervention, in a virtual environment, the number of simulated objects may reach tens. For an acceptable simulation quality, the number of points to describe an object should reach thousands for simple objects and millions for more complex objects. Obviously, in terms of software, concave objects should be divided to subparts (smaller, convex objects).

In view of the situation presented above, it is self-evident that one or more optimizing solutions are required to reduce the number of calculations.

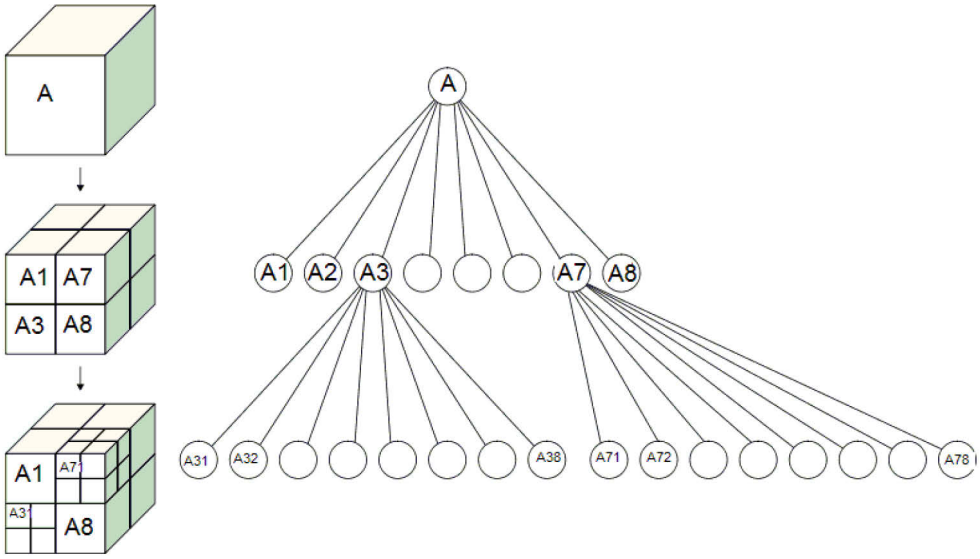
One of the most used algorithms of optimizations is ***Octree*** (Meagher, 1980)

Octree

Octree is a data organizing concept according to which each node or knot, has at most 8 other sub-nodes (children) (Fig. 1).

Visually, if we divide a cube with 3 perpendicular planes at a distance of a/2, where "a" is the main cube's edge, 8 cubes with an "a/2" edge will be obtained. The larger cube is the "parent" of the smaller cubes (children). Dividing cubes will proceed to a certain level (resolution), to a certain eps accuracy ($a_i <$ eps) or to attaining all given points.

Figure 1: Building an octree from root A



Among the established terms and notions we note:

- root = cube with no further parents
- children, branches = subparts of a cube
- parent = the cube which was divided into subparts
- ancestors = the set of cubes which contain a certain subpart
- leafs = the set of subparts which have no other sub-subparts (which do not divide and have no other branches) (Meagher, 1980)

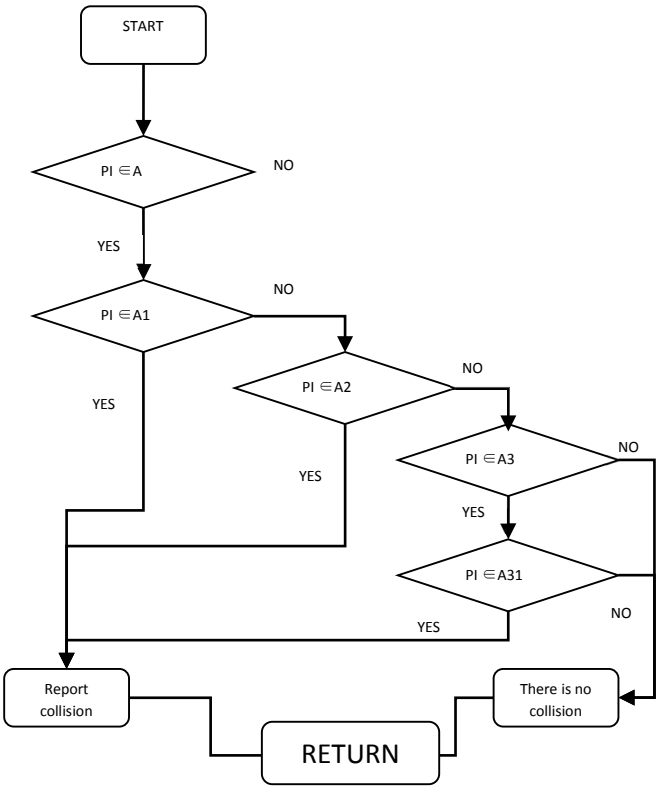
In case of a random object, classic build of an octree starts from the extensive cube (the root) and continues sequentially on each level of branch, assigning points of the "to-be-represented" object to each subpart and thus to the whole structure. To be noticed that the structure may lack cubes if they do not contain any of the points which describe the object. The result will be represented by a list of nodes (points that describe the object) indexed based on a hierarchy.

This type of hierarchy is useful starting with collision detection and ending with deformation calculation (FEM) (Miller, 2007), reaction forces determination, etc. Compared to the AABBs algorithm presented in 3 figure, according to the collision detection algorithm of two objects described by points, one object is stationary (target) and the other is movable (the instrument), implies:

- choosing a point describing the instrument
- testing if that certain point is part of the target's root
- if the point is part of the root, the branches of the root are tested to determine which specific branch contains the point
- the process iterates to the leaf level, specifying exactly where the collision had place

Assuming a target structure such as in Figure 1 and a device with a P1 describing cube A31, then the following tests shall be performed iteratively according to the algorithm above (Fig. 2):

Figure 2: Algorithm AABBs based on octree



The analysis algorithm above notes that the tests for branches A4 to A8 were removed because branch A3 was found valid to the test; tests for branches A32 to A38 were also ruled out because branch A31 was found valid and even more important, testes for children of branches A4 to A8 were also removed.

To be noticed that, naturally, this algorithm is applied to every point of the studied object. Particular, using octree in order to describe objects of surgical virtual reality, the following should be noted:

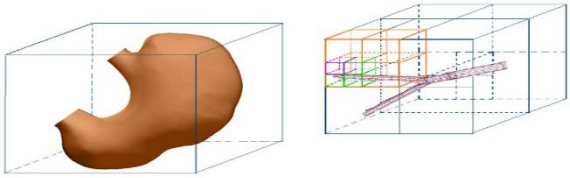
In larger objects cases, objects described by thousands to millions of points which move under the influence of physic parameters or which deform significantly under the influence of mechanical parameters, the octrees should be recalculated after each movement or deformation.

Reconstruction of octrees implies, obviously, an increasing amount of calculations. In case of instruments described by hundreds to thousands of points, applying an algorithm to detect collision between a large object and that specific instrument implies, actually, applying the algorithm for each point describing the instrument.

Double - octree

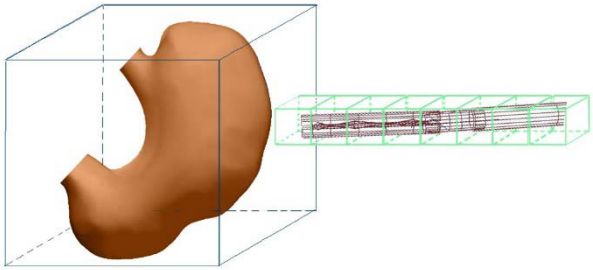
Double-octree algorithm consists in applying the collision detection algorithm between object A (target) and object B (instrument) not by testing progressively object A's leafs compared to all points of object B, but compared progressively with object B's octree (Fig. 3).

Figure 3: Collision detection



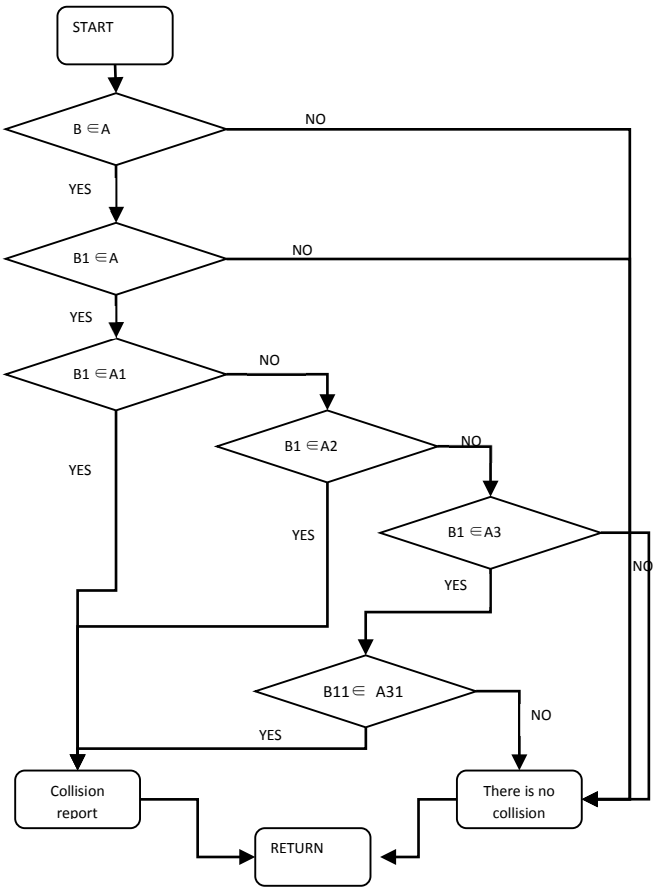
Moreover, in laparoscopic virtual reality, instruments have a specific property: “They do not actually fit inside a cube!” because in most of cases they are long and thin. The structure’s tree building principle in this case is not spatial distribution, but position along the axis of the instrument. Therefore, for instruments it is suggested a structure formed by a set of root-cubes (slices) which divide the instrument into a number of parts of the order of tens (Fig. 4).

Figure 4: Collision detection with an instrument formed by a set of root-cubes



Indexing root-cubes that divide an instrument into slices will be done according to the most likely place of collision (from apex to handle). Assigning object points to the slices, being a characteristic of that instrument, will be executed during the programming. Each octree’s edge values are, obviously, different after the instrument has been moved (rotated, translated). But if assigning points has already been done, recalculating Xmin, Xmax, Ymin, Ymax, Zmin and Zmax only for the cube used for the test, will reduce the required time to fulfill these calculations because the other slices will not be tested. The efficiency of the algorithm will significantly increase, ruling out 90% of the previously executed calculations for instruments with at least 10 root-cubes and 10.000 points (Fig. 5).

Figure 5: Double-octree algorithm



3. Conclusions:

Living in a society in which fighting disease is on a daily agenda, having good, well trained doctors, is a necessity not related to comfort but related to hope of life and life quality. To achieve this, a laparoscopic surgical simulator based on virtual reality is a step forward, it is a priority in Romania.

The new optimization collision detection method based on double-octree algorithm combined with a procedure, which divides an instrument into a set of root-cubes organised as hierarchy at the programming moment reduced volume of calculations by 6 milions times. Elaborating an algorithm described in one page takes a large amount of time and bringing it to an end can not be done without a glow of mind. Using the dual-octree algorithm presented in this article in place of octree algorithm should reduce the number of necessary calculations during the execution of the program, to less than one third!

Dual-octree algorithm can be applied not only in informatics but as well as in any scientific field working with hierarchical data (such as in statistics, etc).

4. References

- Berkley, J., Turkiyyah, G., Berg, D., et al., (2004) Real-Time Finite Element Modeling for Surgery Simulation: An Application to Virtual Suturing. *IEEE Transactions on Visualization and Computer Graphics*, 10(3): p. 314-325.
- Cavusoglu, M.C., Goktekin, T.G., Tendick, F., et al., (2004) GiPSi: An Open Source/Open Architecture Software Development Framework for Surgical Simulation, in Medicine Meets Virtual Reality 12, Newport Beach, CA. p. 46–48.
- Cochior, D., Custură-Crăciun, D., Ciobanu-Oprea, C. (2005) *Realitatea virtuală chirurgicală – utilă sau necesară?*. Editura Electra, p. 167, ISBN 973-7728-24-6. Monograph.
- Coumans E., (2010) Collision Detection and Rigid Body Dynamics. Available at http://graphics.stanford.
- Custură-Crăciun D., Cochior D., Tecuceanu V., Ciobanu-Oprea C., (2004-2006) SIVILAP-Simulator de chirurgie laparoscopică (hardware, software) (PNCDI): Program National INFOSOC SIVILAP.
- El Saddik, A., Orozco, M., Eid, M., Cha, J. (2011) *Haptic technologies bringing touch to multimedia.* Hardcover, p. 218, ISBN: 978-3-642-22657-1. Monograph.
- Ericson, C., (2005) *Real-time Collision Detection*, Morgan Kaufmann.
- Gilbert, E., Johnson, D., and Keerthi, S., (1988) A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space. *IEEE Journal on Robotics and Automation*. 4(2): p. 193-203.
- James, D.L. and Pai, D.K. ArtDefo (1999) Accurate Real Time Deformable Object in The 26th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA: ACM.
- Masutani, Y., Inoue, Y., Ishii, K., et al. (2004) Development of Surgical Simulator Based on FEM and Deformable Volume-Rendering in Medical Imaging, San Diego, CA: The International Society for Optical Engineering.
- Meagher D., (1980) *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*, Rensselaer Polytechnic Institute (Technical Report IPL-TR-80-111).
- Miller, K., Joldes, G., Lance, D., et al., (2007) Total Lagrangian Explicit Dynamics Finite Element Algorithm for Computing Soft Tissue Deformation. *Communications in Numerical Methods in Engineering*. 23(2): p. 121–134.
- Zachmann Gabriel, (2000) *Virtual Reality in Assembly Simulation —Collision Detection, Simulation Algorithms and Interaction Techniques*, Phd Thesis, Dem Fachbereich Informatik der Technischen Universität Darmstadt.
- Zhou, Y., Chen, W., and Tang, Z., (1995) An Elaborate Ambiguity Detection Method for Constructing Isosurfaces within Tetrahedral Meshes, *Computers & Graphics*, 19(3): p. 355-364.