

A technique for optimised navigation in regular geometries

Pedro Arce, John Apostolakis and Gabriele Cosmo

Abstract—The simulation of a DICOM file describing the geometry of a patient through a 3-dimensional grid of several million voxels represents a big challenge in terms of time and memory consumption. We have developed a fast technique to navigate in these regular voxelised geometries in the GEANT4 framework. It takes advantage of the regular structure of the geometry to optimise the location of voxels at tracking time. An option to skip on the fly the voxel boundaries when two contiguous voxels share the same material reduces the number of steps and therefore the simulation time. When the number of materials in the phantom is of the order a few dozens, enough to reach the required precision in medical physics applications, this technique is several times faster than the optimised navigation algorithms implemented in GEANT4, while keeping optimal memory and initialisation time.

A. Description of the problem

In medical physics simulation it is frequent the need to simulate a DICOM image that contains the image of a patient or a test phantom. Those images consists on a big number of voxels, in the order or millions or tens of millions. Usually all voxels have the same dimensions and are touching each other, so that they form what can be considered as a prism with no holes, regularly divided in the three dimensions. Each of these voxel may have a different material. To speed up the simulation, usually the many distinct materials in a human or animal body are grouped in a few. The number of materials depends on the required precision. For radiotherapy applications Plessis et al. recommend using 7 materials[1], while Schneider et al. recommend using 24 materials [2].

B. Navigation techniques in GEANT4

The main challenges for a particle tracking algorithm in a 3D geometry of solid models are: 1) Locating in what solid volume the particles position is; 2) Computing the distance to the closest intersection with a volume boundary along a given trajectory. For a general geometrical structure the CPU time of a naive algorithm scales linearly with the number of solid models to be searched. Different improved algorithms scale logarithmically by optimizing the set of solid models intersected. Some divide the geometrical space in a non-uniform grid, and others utilising an auxiliary tree structure. Uniform grids offer the advantage of very fast searching time (few floating point operations) for locating the volume at a particles position. However normally they have large

memory requirements and may imply unnecessary tracking steps to propagate the particle through the grid boundaries. Both Geant3 [3] and the Geant4 [4] simulation toolkit contain algorithms to address tracking speed and memory use, by means of a generalized uniform grid logic [5] Every volume containing other solid objects is virtually subdivided into grid cells. For each cell a data structure is filled listing the volumes IDs intersecting this cell. In Geant4, if the number of such volumes is greater than a set number (typically 2 or 3), then the grid is refined along one of the remaining orthogonal spatial dimensions, recursively. Neighbouring grid cells with identical lists are equivalent, pointing to the same volumes. Hence the redundant data structures are discarded and these grid cells are treated as a single extended cell. In this way the grid granularity adapts to the 3D geometrical structure but it retains its uniformity. At tracking time it is therefore straightforward to find in which grid cell the particle position is, and only the volumes in its list need to be checked. Geant4 offers several ways to describe a geometry, and for each one a corresponding technique for navigation is utilised. A number of these ways can be utilised to describe regular phantom geometries. The existing descriptions and corresponding navigation techniques did not feature or exploit the regularity which is present in such geometries. In the first description the user will model the phantom geometry by defining a parameterisation, which describes the location and material of a phantom voxel given an integer index. At initialisation the navigation module will prepare a virtual division of the geometry in 1 or 3 dimensions (the number of dimensions is a user choice)[6].. At tracking time Geant4 applies a navigation technique as described in [7]. phantom voxel typically has many candidates, and will incur considerable CPU time. We will call these voxel 1D navigation and voxel 3D navigation. An alternative option provided in Geant4 is to use nested parameterisations to define the phantom image. In this way the search of voxels is done in one dimension at a time, reaching a CPU speed similar to the one using voxel 3D navigation while keeping memory consumption similar to the one characterising the voxel 1D navigation. We will call this nested navigation.

C. Past studies for optimising GEANT4 navigation in regular geometries

Even with the above mentioned techniques, the simulation in DICOM files takes a big amount of CPU time for a usual DICOM file, where the number of voxels is of the order of 10 million. Several attempts have been made to optimise navigation in regular geometries; we briefly cite them here.

P. Arce is with the Department of Basic Research, CIEMAT, Madrid, Spain, e-mail:pedro.arce@ciemat.es

J. Apostolakis and G. Cosmo are with the SFT Division, CERN, Geneva, Switzerland

H. Jiang and H. Paganetti[8], and later A. Raijmakers [9] use a fast navigation algorithm by looking at the six nearest voxels only and assigning dynamically the mass density.

K. Sutherland proposes an alternative parameterised navigation algorithm making use of fast voxel location.

V. Huber-Tremblay, L. Archambault, D. Tubic, R. Roy and L. Beaulieu[10] reduce the number of voxels by using the octree method.

D. Sarrut and L. Guigues [11] propose a new segmentation in regions of arbitrary shape to reduce the number of voxels.

I. THE PROPOSED APPROACH

In this approach a specialised parameterisation class *G4PhantomParameterisation* is used to define the regular structure of a phantom geometry. The definition of the voxels uses the minimum set of parameters (dimension, offset and number of voxels in each of the three dimensions); it builds a list of voxel-material indices for fast association with materials. The corresponding, specialised, navigation class, *G4RegularNavigation* uses this parameterisation to directly locate the phantom voxel visited by the tracks in the simulation (optimisation cells or virtual divisions are not created to aid in location or navigation).

An option is provided to skip the boundaries of contiguous phantom voxels when these share the same material. As a result, the total number of steps can be significantly reduced. This is computed on the fly, and there is neither a penalty in initialization time nor an increase in memory consumption for this option. We will call this technique regular navigation.

This approach applied to the case of consecutive voxels of a voxelised phantom geometry extends a key concept used in virtual divisions in Geant3 (and its generalisation, smart voxels in Geant4): to merge a set of cells with the same contents. When the option of skipping boundaries is turned on, it increases the effective size of volumes seen by the tracking to a maximal degree, as does the approach of D. Sarrut et al [11]. This option can be turned off, to be used, for example, in the case when there are so many materials that contiguous voxels rarely share the same material.

When several voxels are traversed in one step, the energy deposited is only calculated once and therefore it is not properly distributed among the voxels. We can easily extract the information of the geometrical step length traversed at each step, but the true step length, that takes into account the multiple scattering, is only calculated once by Geant4. We have simulated the dose deposited in a water phantom by the particles that have traversed a VARIAN 2100 6 MeV accelerator. To this dose we applied a simple correction distributing the dose among the voxels proportionally to the geometrical step length travelled in each voxel. Fig. 1 it is compared the dose calculated with this correction with the dose calculated when the option of skipping voxel frontiers is not activated. As it can be seen, the dose is slightly bigger in the first voxels. This difference is due to a double effect that contribute to a bigger energy lost in the voxels at deeper depth (considering that most of tracks travel in the direction of positive Z): when the particle travels through the voxels it

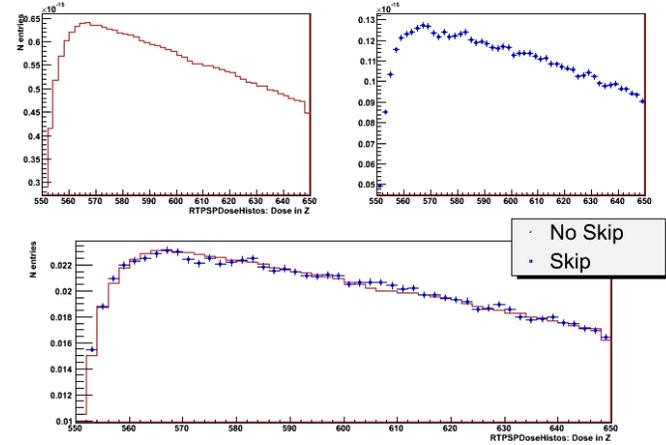


Fig. 1. Comparison of dose deposited in a water phantom for the case when voxel frontiers are not skipped and the case when they are skipped and a simple correction is applied distributing the dose among the voxels proportionally to the geometrical step length in each voxel

loses energy and therefore the energy lost per length ($dEdx$) is bigger and also the effect of the multiple scattering is bigger.

To implement a better correction we cannot calculate in a simple way the energy lost at each voxel, because the step length increase due multiple scattering (that converts the geometrical step length in what we will call the true step length) and the energy loss are correlated. We have therefore implemented an iterative algorithm: we first distribute the total true step length among the voxel proportionally to their geometrical step length; with these values we calculate one voxel after another the value of $dEdx$ and then the value of the kinetic energy at the entrance of each voxel; with these values we calculate the geometrical to true step corrections due to multiple scattering for each voxel; finally we use these new values to recalculate the energy lost in each voxel. Fig. 2 compares the dose deposited in a water phantom by the particles that have traversed a VARIAN 2100 6 MeV accelerator after applying this correction algorithm and the dose calculated when the option of skipping voxel frontiers is not activated. It can be seen that the two-step iteration described is enough to have a very good agreement between both distributions.

The modular design of the Geant4 Geometry Modeler [4], has made it possible to integrate this navigation technique with a relatively small number of modifications

II. RESULTS

We have simulated several voxelised image files corresponding to real CT scans of human body with different number of voxels. Table I presents the results of the different navigation techniques for the case of a phantom of $256 \times 256 \times 68 = 4.5$. The table shows the time in seconds on an IntelCore2 at 2.0 GHz to simulate 1000 geantinos (special particles that have no physics, only transportation) through the full phantom, from one wall towards the centre until the opposite wall in any direction. The seven columns in this table show the calculations with the *voxel 1D navigation*, *voxel 3D navigation*, *nested navigation*, *regular navigation* with 4 materials,

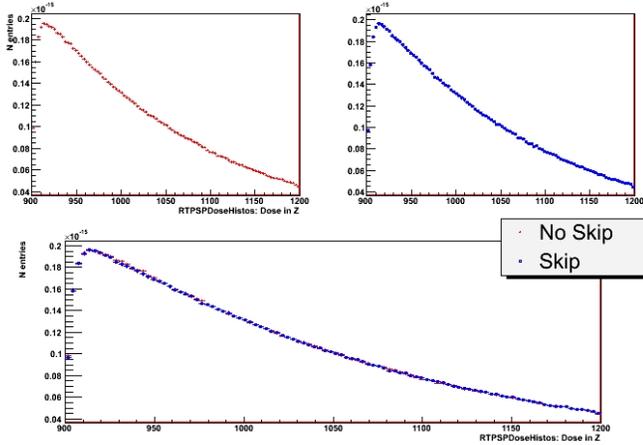


Fig. 2. Comparison of dose deposited in a water phantom for the case when voxel frontiers are not skipped and the case when they are skipped and the correction described in the text is applied for the distribution of dose among the voxels

TABLE I
CPU TIME SPENT ON TRACKING 1000 GEANTINOS (SEC.)

Vx 1-D	Vx 3-D	Nest	Reg(4)	Reg(24)	Reg(250)	Reg(no)
2030.	1.98	1.62	0.30	0.40	0.74	1.45

TABLE II
MEMORY SPENT ON TRACKING 1000 GEANTINOS (MB)

Voxel 1-D	Voxel 3-D	Nested	Regular
199	1348	199	199

TABLE III
INITIALISATION TIME SPENT BEFORE FIRST PARTICLE IS TRACKED (SEC.)

Voxel 1-D	Voxel 3-D	Nested	Regular
55	105	55	55

regular navigation with 25 materials, *regular navigation* with 250 materials, *regular navigation* with no skipping of voxel frontiers. Except where it is indicated 24 materials have been used.

It is clear that skipping frontiers of voxels with same material will reduce the time more when the number of materials is less. Therefore for the regular navigation (the only one affected by this option) we have tried different options, by using different number of materials in the phantom, namely 4, 24 and 250 and also shown the result with no frontier skipping.

Table II shows the memory consumption. Table III shows the initialization time spent until the first particle starts to be tracked.

III. CONCLUSIONS

We are presenting a new navigation technique specialised for regular geometries, which provides improved CPU time in navigation and initialisation and much more efficient use of memory. The new algorithm is fast and efficient since not requiring additional memory or initialization time. The gain in speed is greater the smaller is the number of materials chosen, but even with a number of materials as big as several hundreds

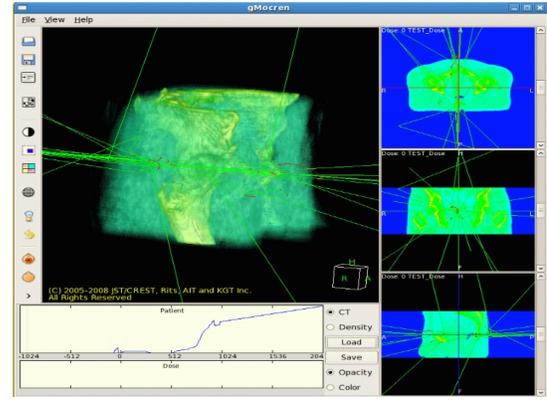


Fig. 3. GEANT4-gMocren [12] visualisation of tracks simulated with the proposed algorithm in a pelvis phantom (photons in green and electrons in red)

(much bigger than what will probably needed to reach the required precision) it is still performing fast.

REFERENCES

- [1] F. C. Du Plessis, C. A. Willemsse, M. G. Lotter, L. Goedhals. *The indirect use of CT numbers to establish material properties needed for Monte Carlo Calculation of dose distributions in patients*, Med. Phys. 25(7):1195-1201 (1998)
- [2] W. Schneider, T. Bortfeld, W. Schlegel. *Correlation between CT numbers and tissue parameters needed for Monte Carlo simulations of clinical dose distributions*, Phys. Med. Biol. 45(2):459-478 (2000)
- [3] CERN Program Library Long Writeup W5013
- [4] Agostinelli S et al., *Geant4 - A simulation toolkit*, Nucl. Instrum. Methods Phys. Res. A 506 250303 (2003)
- [5] S.Giani et al., *GEANT steps into the future*, CHEP94 Proceedings pag.382-384, LBL-35822 Conf-940492 UC-405, USA, 1994
- [6] G. Cosmo. *Modeling detector geometries in Geant4*, Nuclear Science Symposium Conference Record, 2003 IEEE, Volume 1:479-481 (2003)
- [7] P. Kent *Pure tracking and geometry*, CERN Internal Note, April 1995.
- [8] H. Jiang and H. Paganetti. *Adaptation of GEANT4 to Monte Carlo dose calculations based on CT data*, Med. Phys. 31:2811-2818 (2004)
- [9] A.J.E. Raaijmakers Ph.D. Dissertation, Utrecht University (2008)
- [10] V. Huber-Tremblay, L. Archambault, D. Tubic, R. Roy and L. Beaulieu *Ocree indexing of DICOM images for voxel number reduction and for improvement of Monte Carlo simulation computing efficiency*, Med. Phys. 33(8):2819-2831 (2006)
- [11] D. Sarrut and L. Guigues. *Region-oriented CT image representation for reducing computing time of Monte Carlo simulations*, Med. Phys. 35(4):1452-1463 (2008)
- [12] <http://geant4.kek.jp/gMocren>