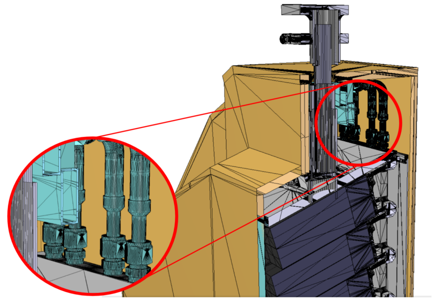
TGeoArbN - a ROOT compatible triangle mesh geometry implementation

Towards a more automated CAD to ROOT conversion through tessellation compatible with ROOT's default geometry package

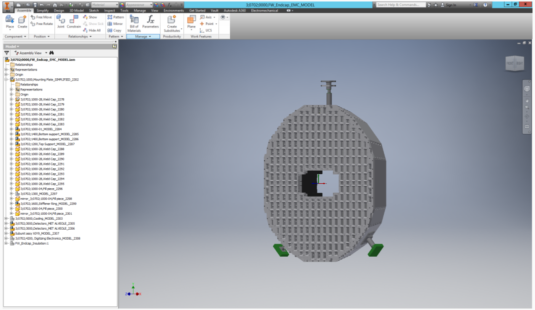
Ben William Salisbury

October 23, 2024

HISKP University Bonn



From CAD to simulatable ROOT-Geometries for $\overline{\text{P}}$ANDA



- The detector designs of $\overline{\text{P}}$ANDA's Forward Endcap outpaced its corresponding **ROOT** simulation geometry
- We want a way of converting CAD into simulatable ROOT geometries
- We do not want to have to manually recreate complex shapes out of ROOT's primitive geometries
- We are not too concerned about simulation time (at the moment)

Our requirements for an automated conversion from CAD to ROOT

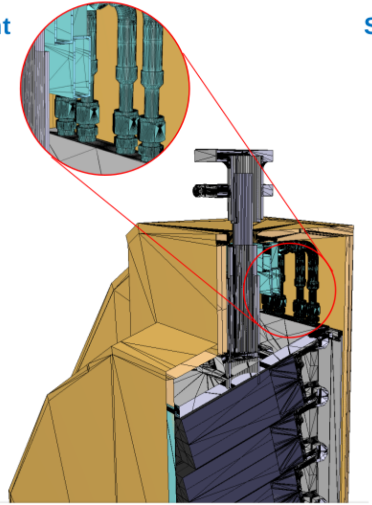
We need to deal with:

Complex CAD Component

- complex structure
- Not mappable to a ROOT geometry (box, tube,...)

(triangle) mesh tessellation

TGeoArbN
(root geo. object)



root geo. of PANDA forward endcap after automatic import

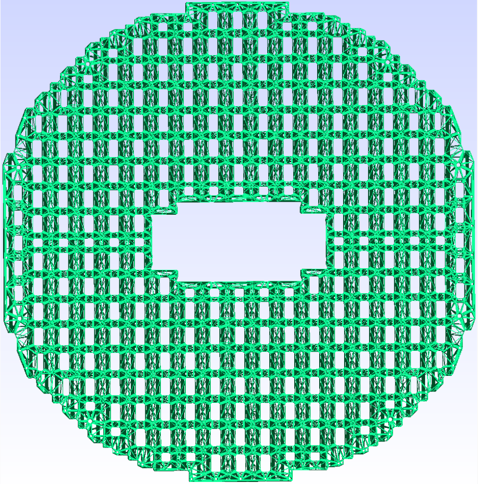
Subdetector Model

- consist of many components
- many positions and orientations

export STEP files

CAD-Converter
(create root geo. volumes)

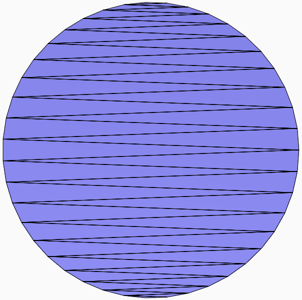
TGeoArbN - ansatz



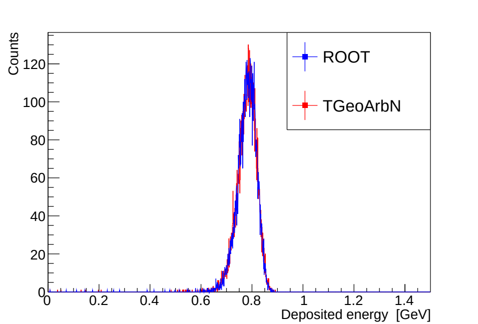
- Aim of using TGeoArbN in VMC simulations requires to implement ROOT's TGeoShape interface
- Generic **triangle mesh** requires generic implementation approach for TGeoShape methods: **ray casting**
- **Triangle mesh** + **ray casting** → simulatable geometry comparable to G4TessellatedSolid in geant4

TGeoArbN: Triangle mesh

Triangle meshes are fantastically flexible, but can only approximate curves. This does lead to differences for example for cylinders resembled by TGeoArbN and TGeoTube.



(a) Tessellated base of a cylinder.

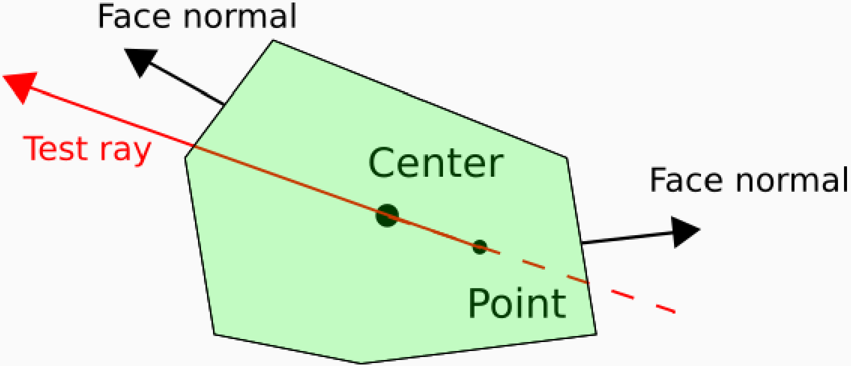


(b) Deposited energy of a 1 GeV photon in a cylindric form.

Deviations depend on **triangle mesh**-export and geometry

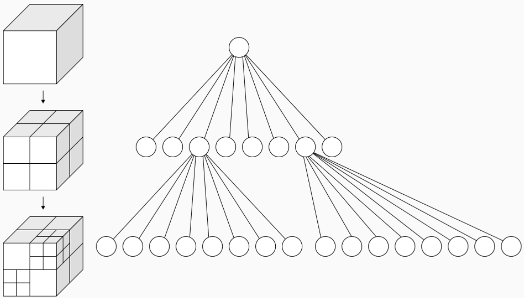
TGeoArbN: ray casting - Contains() implementation

Contains(): Is a point (particle) contained by the geometry?



Got to test **each** triangle for an intersection with the test ray! Horrible scaling!

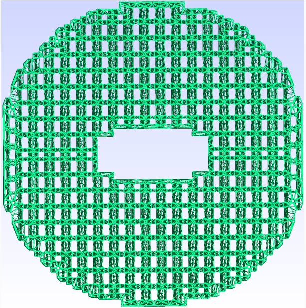
TGeoArbN: Scaling issue - Apply partitioning structure

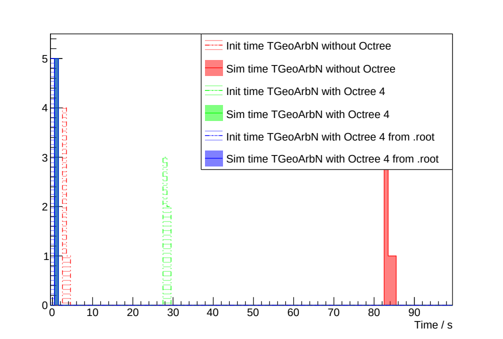


Octree-concept: picture taken from *WhiteTimberwolf 2010*

- Bounding box recursively split into eight identical sub-boxes (octants)
- Splitting occurs if a box contains more than x triangles
- Test triangles in relevant boxes $\rightarrow$ **can** yield benefits in simulation time, but can be costly

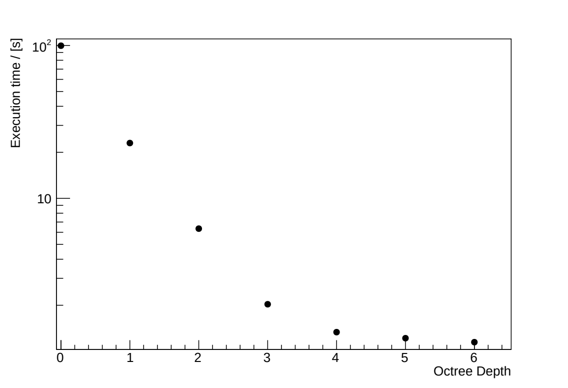
TGeoArbN: Simulation time - Effects of an Octree





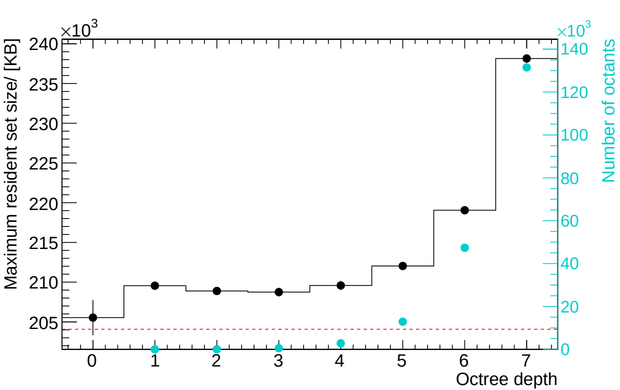
- Simulation time for 10 1 GeV/c photons shot at lead tungstate Backplate (consisting of 277k triangles!)
- Using (and pre-building) an Octree can be very beneficial!

TGeoArbN: Using an Octree - Effects of different depths



- Simulation time for 10 1 GeV/c photons shot at lead tungstate Backplate (consisting of 277k triangles!) !!done on other machine!!
- Benefits level off for increasing Octree depth for the Backplate.

TGeoArbN: Using an Octree - Memory Usage

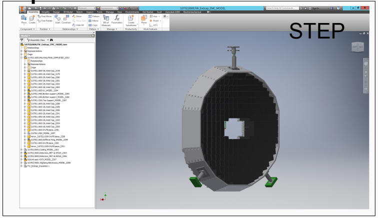


Simulation application memory usage for a 598 triangle mesh forming a cylinder for different octree depths, compared to use of TGeoTube (red line).

- Simulation time decrease comes at the cost of memory!

CADtoROOT-Converter (based on T.*Stockmanns 2012*)

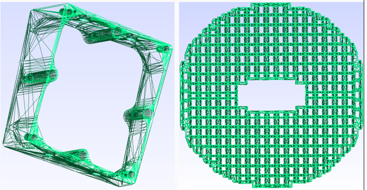
Input:

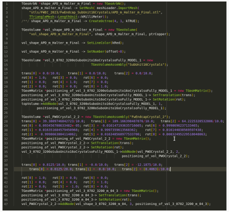


CAD-Converter
- T. Stockmanns' auto-recognition
- our automatic tessellation

CAD-Converter output:

STLs/Meshes



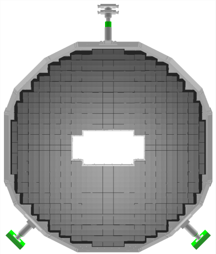


Geometry defining ROOT script

Additional ROOT script building geometry and assigning materials

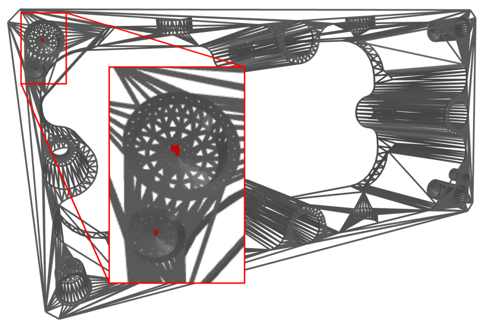
Final output:

.root file containing geometry as well as material assignment



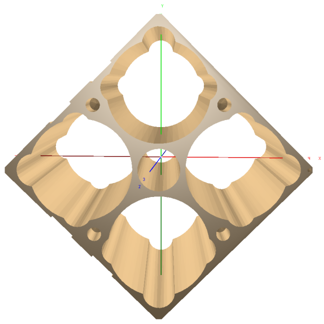
Automatic conversion chain - Remaining problems

- Some meshes are faulty (partly due to CAD itself, partly due to tessellation)!



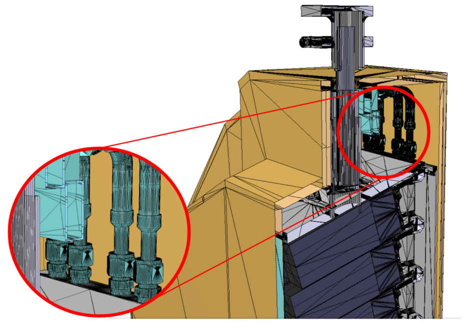
Automatic conversion chain - Remaining problems

- Some meshes are faulty (partly due to CAD itself, partly due to tessellation)!
- Meshes produced in non-ideal reference frame lead to non optimal bounding boxes!



Automatic conversion chain - Remaining problems

- Some meshes are faulty (partly due to CAD itself, partly due to tessellation)!
- Meshes produced in non-ideal reference frame lead to non optimal bounding boxes!
- Geometry relies heavily on `TGeoArbN` $\rightarrow$ performance is an issue! However, the partitioning structure helps a lot with the more complex meshes!



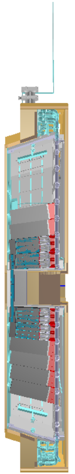
Summary and outlook

Summary

- Triangle mesh geometry compatible with ROOT's default geometry package
- Automated geometry conversion from CAD to ROOT (used to convert $\overline{\text{P}}$ANDA's FwEndcap geometry)

Outlook

S. Neuhaus from Wuppertal University (presenting a poster) started comparing `TGeoArbN` with ROOT's VecGeom tessellation implementation

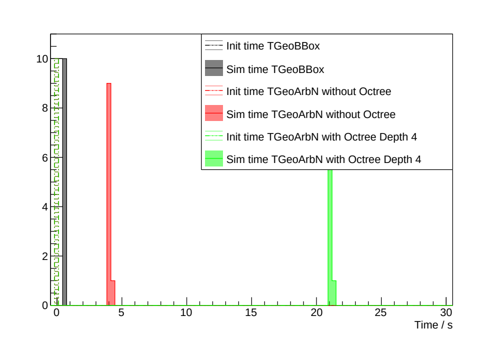


Bibliography

- Stockmanns, Tobias (Dec. 2012). "STEP-to-ROOT – from CAD to Monte Carlo Simulation". In: *Journal of Physics: Conference Series* 396.2, p. 022050. DOI: 10.1088/1742-6596/396/2/022050. URL: https://doi.org/10.1088/1742-6596/396/2/022050.

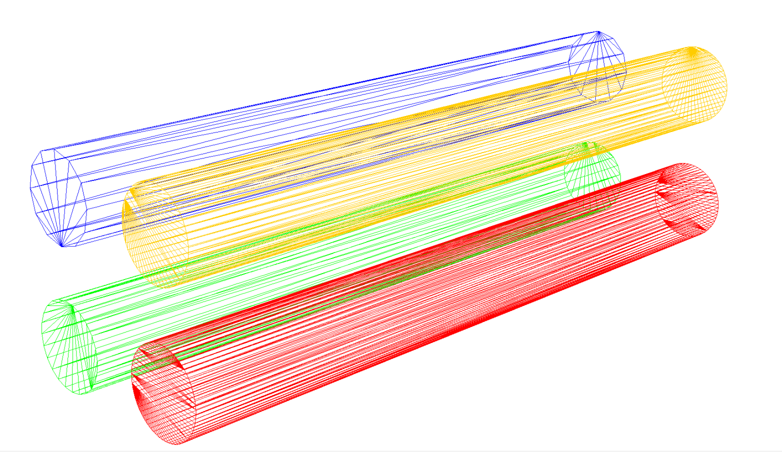
  WhiteTimberwolf (Mar. 27, 2010). *Schematic drawing of an octree, a data structure of computer science.* URL: https://commons.wikimedia.org/wiki/File:Octree2.svg (visited on 10/15/2024).

Appendix - Octree problems

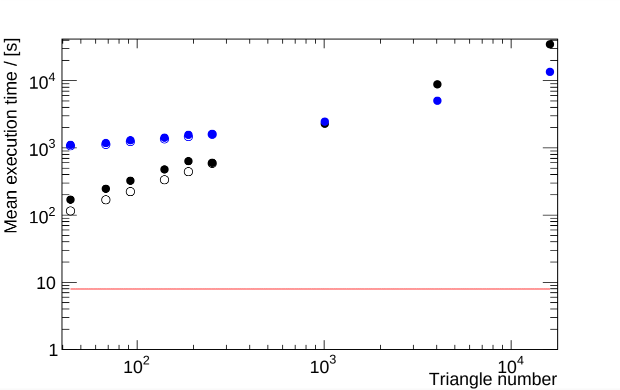


- Simulation time for 1000 1 GeV/c photons shot at lead tungstate box
- Octree induced overhead increases simulation time for smaller meshes!

Appendix - Cylinders



Different cylinder meshes with increasing triangle count



Simulation time increase for triangle meshes forming cylinders for increasing triangle counts

black: without Octree, blue: with Octree of depth 4, red: TGeoTube