Recent Updates in USOLIDS/VecGeom (focus on ALICE requirements)

Sandro Wenzel / CERN-ALICE

for the VecGeom dev-team

Geant4 collaboration meeting, Ferrara, 15.09.2015

Reminder of Motivation

Incl.	Self	Called	Function	Location
9.63	0.54	83 953 762	TGeoShapeAssembly::Safety(doubl…	libGeom.so.5.34.30: TGeoShapeAss…
6.24	0.14	54 409 322	TGeoShapeAssembly::DistFromOut…	libGeom.so.5.34.30: TGeoShapeAss…
4.05	0.15	56 657 325	TGeoShapeAssembly::DistFromOut…	libGeom.so.5.34.30: TGeoShapeAss…
2.42	0.23	858 580 402	TGeoVoxelFinder::GetNextVoxel(do…	libGeom.so.5.34.30: TGeoVoxelFind…
2.20	1.75	740 115 706	TGeoVoxelFinder::GetNextCandidat…	libGeom.so.5.34.30: TGeoVoxelFind…
1.81	0.28	704 753 410	TGeoXtru::Contains(double const*) …	libGeom.so.5.34.30: TGeoXtru.cxx, …
1.70	0.07	81 244 037	TGeoShapeAssembly::Contains(do…	libGeom.so.5.34.30: TGeoShapeAss…
1.29	0.05	73 509 785	TGeoSubtraction::Safety(double co…	libGeom.so.5.34.30: TGeoBoolNode…
1.24	0.45	309 067 908	TGeoVoxelFinder::GetCheckList(do…	libGeom.so.5.34.30: TGeoVoxelFind…
1.07	0.04	46 839 574	TGeoShapeAssembly::Contains(do…	libGeom.so.5.34.30: TGeoShapeAss…
0.84	0.05	75 659 866	TGeoSubtraction::Safety(double co…	libGeom.so.5.34.30: TGeoBoolNode…
0.81	0.06	136 620 576	TGeoShape::SafetyPhi(double cons…	libGeom.so.5.34.30: TGeoShape.cx…
0.81	0.76	2 332 685 480	TGeoTranslation::MasterToLocal(do…	libGeom.so.5.34.30: TGeoMatrix.cxx
0.79	0.77	199 772 850	TGeoShapeAssembly::Safety(doubl…	libGeom.so.5.34.30: TGeoShapeAss…
0.79	0.25	118 464 696	TGeoVoxelFinder::SortCrossedVox…	libGeom.so.5.34.30: TGeoVoxelFind…
0.76	0.19	1 439 414 508	TGeoXtru::GetThreadData() const	libGeom.so.5.34.30: TGeoXtru.cxx
0.71	0.18	194 586 867	TGeoXtru::DistToPlane(double cons…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.68	0.04	14 337 390	TGeoXtru::DistFromOutside(double …	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.56	0.06	192 608 734	TGeoXtru::Safety(double const*, bo…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.54	0.34	374 978 915	TGeoXtru::GetPlaneVertices(int, int,…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.45	0.08	8 005 726	TGeoXtru::SafetyToSector(double c…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.44	0.06	116 547 273	TGeoXtru::SetCurrentZ(double, int)	libGeom.so.5.34.30: TGeoXtru.cxx
0.43	0.06	76 043 241	TGeoTubeSeg::Safety(double const…	libGeom.so.5.34.30: TGeoTube.cxx, …
0.40	0.33	125 157 334	TGeoXtru::SetCurrentVertices(doubl…	libGeom.so.5.34.30: TGeoXtru.cxx

- Lots of CPU cycles spent in geometry/solid functions
  - at least true for most "intensity frontier" experiments/simulations
  - here shown for ALICE (using TGeo) in a realistic Pb-Pb collision simulation but equally true for other experiments using Geant4 geometry

Reminder of Motivation

Incl.	Self	Called	Function	Location
9.63	0.54	83 953 762	TGeoShapeAssembly::Safety(doubl…	libGeom.so.5.34.30: TGeoShapeAss…
6.24	0.14	54 409 322	TGeoShapeAssembly::DistFromOut…	libGeom.so.5.34.30: TGeoShapeAss…
4.05	0.15	56 657 325	TGeoShapeAssembly::DistFromOut…	libGeom.so.5.34.30: TGeoShapeAss…
2.42	0.23	858 580 402	TGeoVoxelFinder::GetNextVoxel(do…	libGeom.so.5.34.30: TGeoVoxelFind…
2.20	1.75	740 115 706	TGeoVoxelFinder::GetNextCandidat…	libGeom.so.5.34.30: TGeoVoxelFind…
1.81	0.28	704 753 410	TGeoXtru::Contains(double const*) …	libGeom.so.5.34.30: TGeoXtru.cxx, …
1.70	0.07	81 244 037	TGeoShapeAssembly::Contains(do…	libGeom.so.5.34.30: TGeoShapeAss…
1.29	0.05	73 509 785	TGeoSubtraction::Safety(double co…	libGeom.so.5.34.30: TGeoBoolNode…
1.24	0.45	309 067 908	TGeoVoxelFinder::GetCheckList(do…	libGeom.so.5.34.30: TGeoVoxelFind…
1.07	0.04	46 839 574	TGeoShapeAssembly::Contains(do…	libGeom.so.5.34.30: TGeoShapeAss…
0.84	0.05	75 659 866	TGeoSubtraction::Safety(double co…	libGeom.so.5.34.30: TGeoBoolNode…
0.81	0.06	136 620 576	TGeoShape::SafetyPhi(double cons…	libGeom.so.5.34.30: TGeoShape.cx…
0.81	0.76	2 332 685 480	TGeoTranslation::MasterToLocal(do…	libGeom.so.5.34.30: TGeoMatrix.cxx
0.79	0.77	199 772 850	TGeoShapeAssembly::Safety(doubl…	libGeom.so.5.34.30: TGeoShapeAss…
0.79	0.25	118 464 696	TGeoVoxelFinder::SortCrossedVox…	libGeom.so.5.34.30: TGeoVoxelFind…
0.76	0.19	1 439 414 508	TGeoXtru::GetThreadData() const	libGeom.so.5.34.30: TGeoXtru.cxx
0.71	0.18	194 586 867	TGeoXtru::DistToPlane(double cons…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.68	0.04	14 337 390	TGeoXtru::DistFromOutside(double …	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.56	0.06	192 608 734	TGeoXtru::Safety(double const*, bo…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.54	0.34	374 978 915	TGeoXtru::GetPlaneVertices(int, int,…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.45	0.08	8 005 726	TGeoXtru::SafetyToSector(double c…	libGeom.so.5.34.30: TGeoXtru.cxx, …
0.44	0.06	116 547 273	TGeoXtru::SetCurrentZ(double, int)	libGeom.so.5.34.30: TGeoXtru.cxx
0.43	0.06	76 043 241	TGeoTubeSeg::Safety(double const…	libGeom.so.5.34.30: TGeoTube.cxx, …
0.40	0.33	125 157 334	TGeoXtru::SetCurrentVertices(doubl…	libGeom.so.5.34.30: TGeoXtru.cxx

- Lots of CPU cycles spent in geometry/solid functions
  - at least true for most "intensity frontier" experiments/simulations
  - here shown for ALICE (using TGeo) in a realistic Pb-Pb collision simulation but equally true for other experiments using Geant4 geometry
- **Unified Solids (USOLIDS)** project was launched ~2010 to "improve speed/ algorithms/code/maintenance burden" of geometry code for the benefit of Geant4/TGeo…

Reminder VecGeom project

VecGeom = Unified Solids Project

+ Many-Particle API

+ Geometry Model / Navigation

Vec Geom = SIMD/GPU support complete geometry modeler

hosted at gitlab.cern.ch/VecGeom/VecGeom

Short outline

- Comprehensive overview of USOLIDS/VecGeom presented during the last meeting(s)

- **Focus today** on two important requirements for ALICE

- see Gabriele's talks on USOLIDS/G4 integration

Requirements from ALICE

- USOLIDS currently offers implementation to satisfy requirements of CMS, LHCB and others; (see Gabriele's talk for status report on integration)
- ALICE expressed interest; take a look at most important CPU consumers:

Solid	Safety	DistToIn	DistToOut	Contains	Sum	
Pgon	2.05	2.52	0.18	1.18	**5.93**	available in optimized form in USOLIDS ←
Xtru	0.60	0.68	0.20	1.81	**3.29**	
Pcon	1.07	0.32	0.05	0.13	**1.57**	available in optimized form in USOLIDS ←
Assembly*	10.5*	6.24*	6*	2.7*	**23.49***	

numbers represent total percentage cost of simulation runtime

*not elementary solid / inclusive cost

Requirements from ALICE

- USOLIDS currently offers implementation to satisfy requirements of CMS, LHCB and others; (see Gabriele's talk for status report on integration)
- ALICE expressed interest; take a look at most important CPU consumers:

Solid	Safety	DistToIn	DistToOut	Contains	Sum	
Pgon	2.05	2.52	0.18	1.18	**5.93**	available in optimized form in USOLIDS ←
Xtru	0.60	0.68	0.20	1.81	**3.29**	← no dedicated implementation
Pcon	1.07	0.32	0.05	0.13	**1.57**	available in optimized form in USOLIDS ←
Assembly*	10.5*	6.24*	6*	2.7*	**23.49***	

numbers represent total percentage cost of simulation runtime

*not elementary solid / inclusive cost

Requirements from ALICE

- USOLIDS currently offers implementation to satisfy requirements of CMS, LHCB and others; (see Gabriele's talk for status report on integration)
- ALICE expressed interest; take a look at most important CPU consumers:

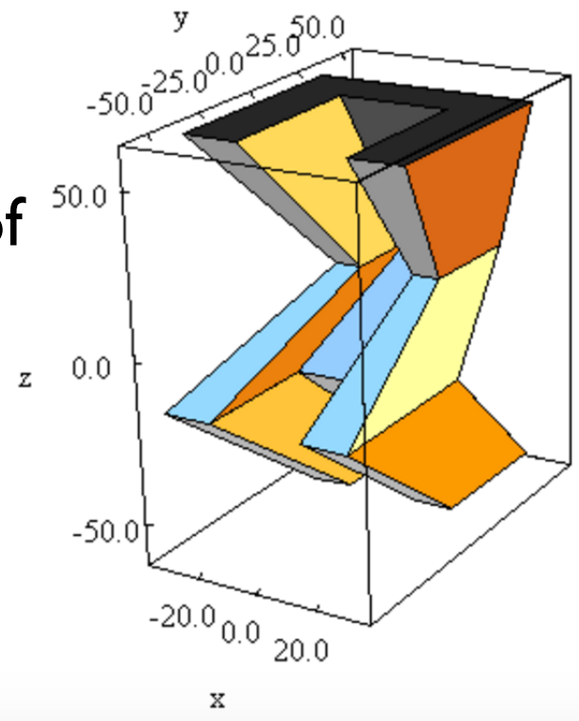
Solid	Safety	DistToIn	DistToOut	Contains	Sum	
Pgon	2.05	2.52	0.18	1.18	**5.93**	available in optimized form in USOLIDS ←
Xtru	0.60	0.68	0.20	1.81	**3.29**	← no dedicated implementation
Pcon	1.07	0.32	0.05	0.13	**1.57**	available in optimized form in USOLIDS ←
Assembly*	10.5*	6.24*	6*	2.7*	**23.49***	← important navigation feature; not implemented

numbers represent total percentage cost of simulation runtime

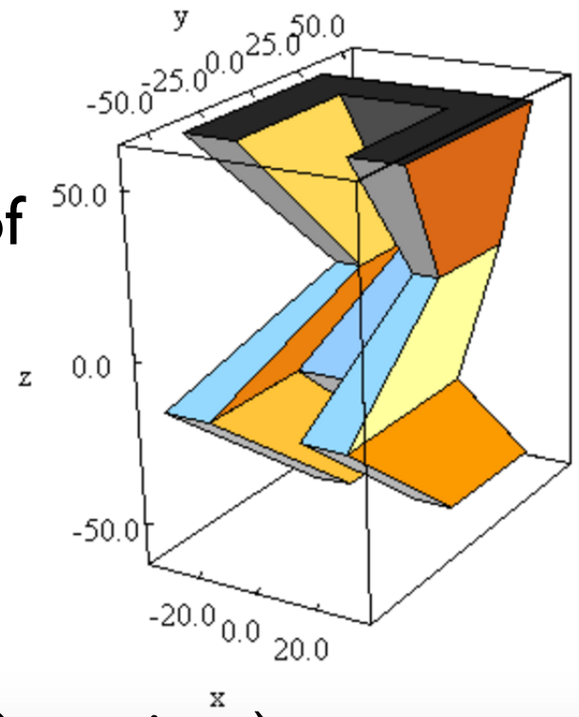
*not elementary solid / inclusive cost

The (simple) Extruded Solid

- Extruded solid is a 2D concave/convex polygonal template extruded along z direction (under application of scale/shifts)
- quite a complex solid

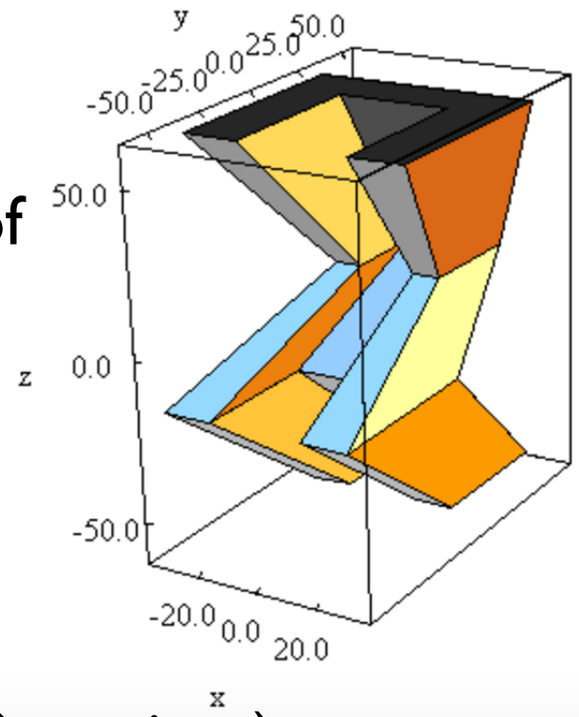


The (simple) Extruded Solid



- Extruded solid is a 2D concave/convex polygonal template extruded along z direction (under application of scale/shifts)
- quite a complex solid
- analysis of ALICE detector showed:
  - ~190 different extruded solids used (from 4 to ~100 vertices)
  - **all** of them consist just of 2 z-planes
  - **none** of them uses scale/transform

The (simple) Extruded Solid



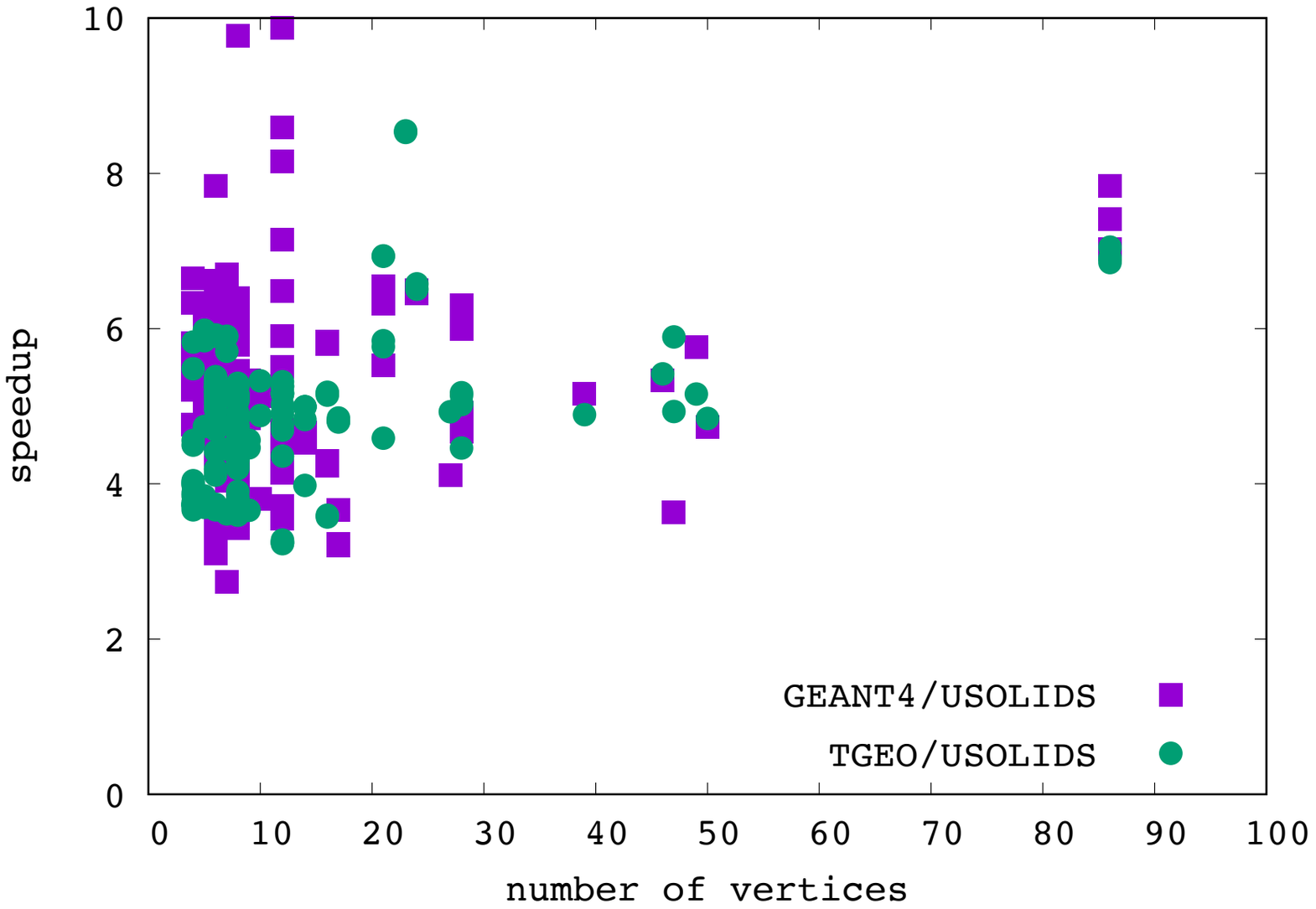
- Extruded solid is a 2D concave/convex polygonal template extruded along z direction (under application of scale/shifts)
- quite a complex solid
- analysis of ALICE detector showed:
  - ~190 different extruded solids used (from 4 to ~100 vertices)
  - **all** of them consist just of 2 z-planes
  - **none** of them uses scale/transform
- Appropriate to provide specialized solid for this use-case
  - The "simple extruded solid" (SExtru)??
  - "Polygonal prism" might be more appropriate name …

The (simple) Extruded Solid

- An implementation of SExtru is now available in the master branch of VecGeom
  - implementation using modern C++; multi-particle interface; GPU ready

The (simple) Extruded Solid

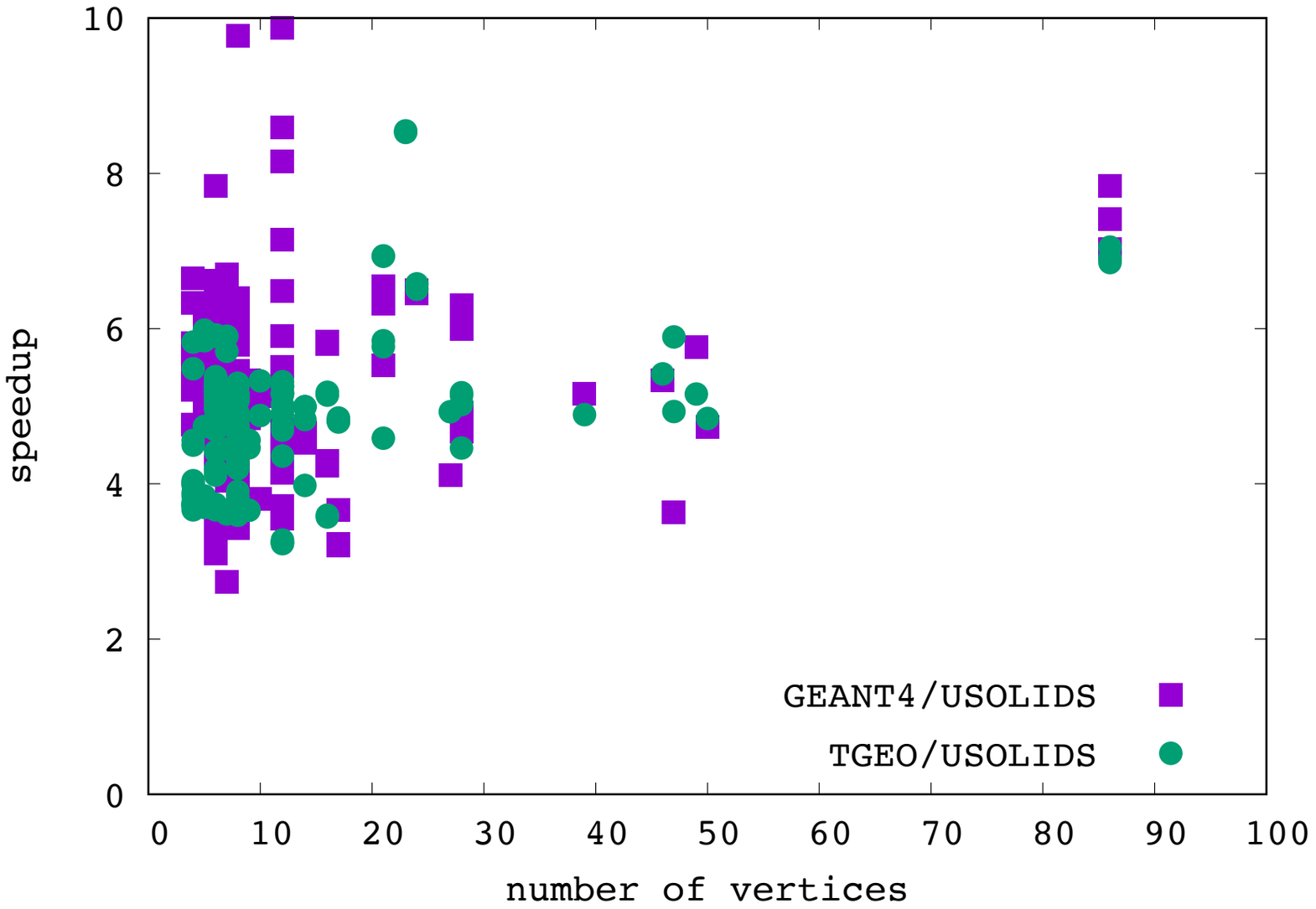
- An implementation of SExtru is now available in the master branch of VecGeom
  - implementation using modern C++; multi-particle interface; GPU ready
- First performance evaluations taking into account all ALICE realizations



Speedup for DistanceToIn

The (simple) Extruded Solid

- An implementation of SExtru is now available in the master branch of VecGeom
  - implementation using modern C++; multi-particle interface; GPU ready
- First performance evaluations taking into account all ALICE realizations

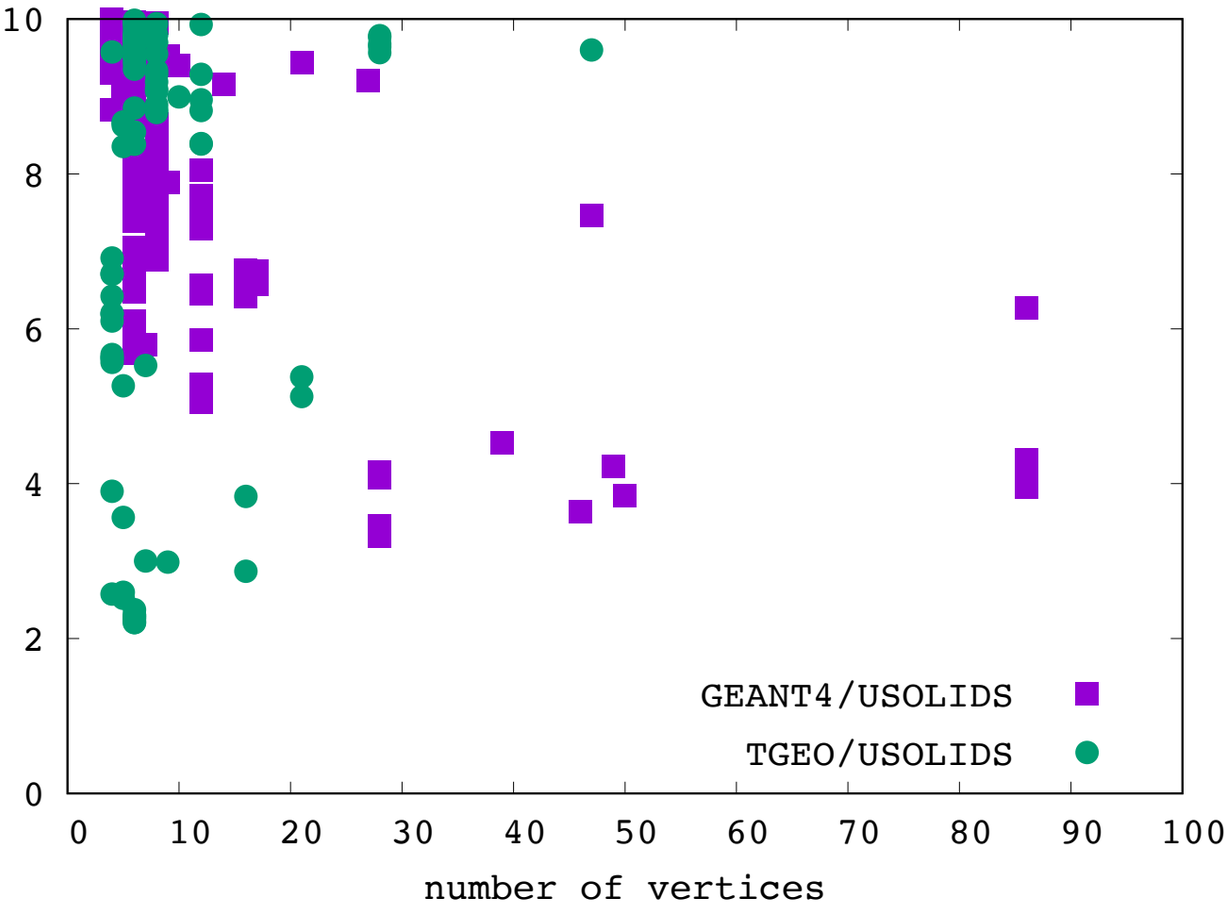


Speedup for DistanceToIn

mean gain **~5ish** over existing G4/TGeo implementations

SExtru Performance (2)

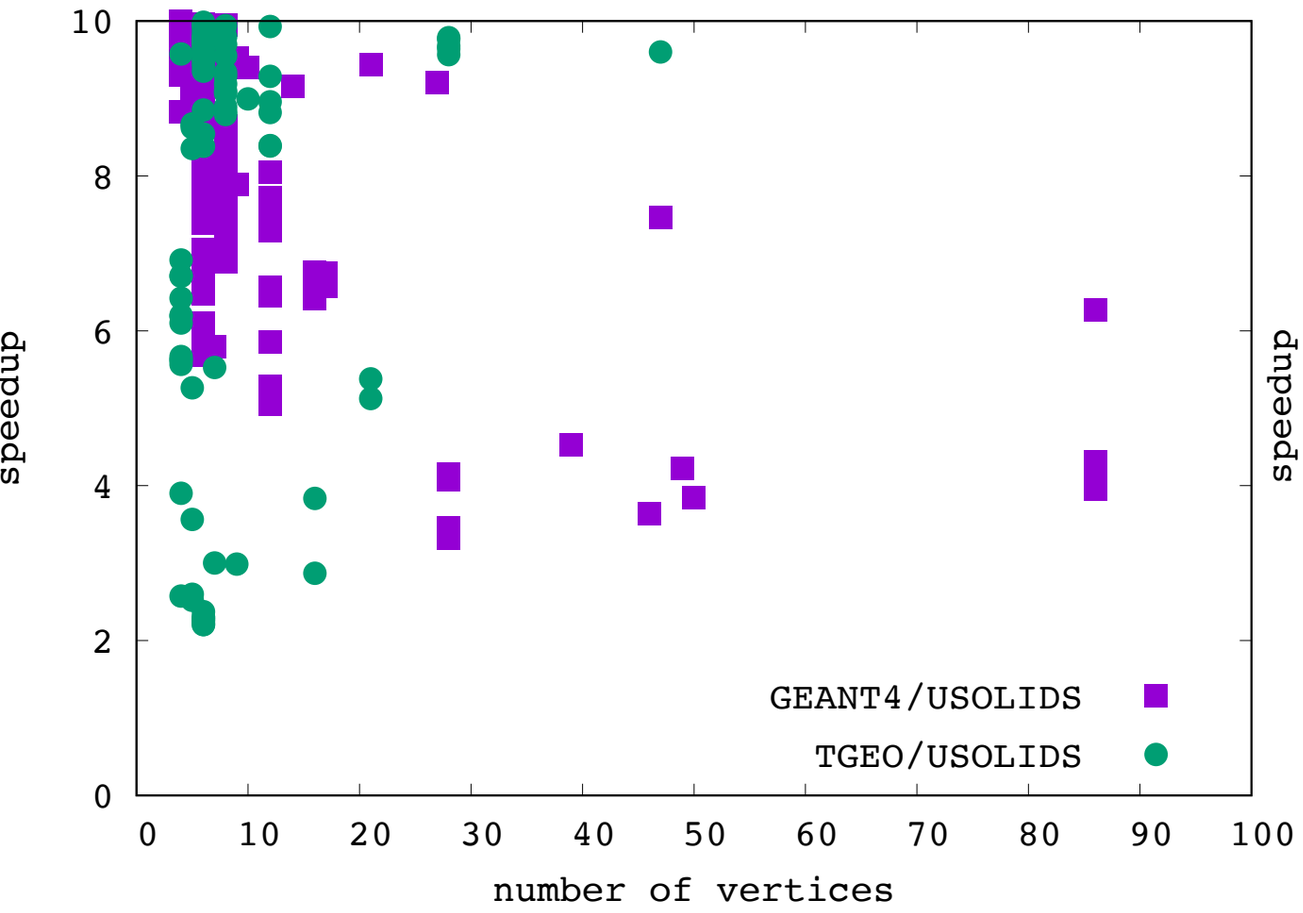
Speedup DistanceToOut



mean gain >**5ish** over existing G4/TGeo implementations

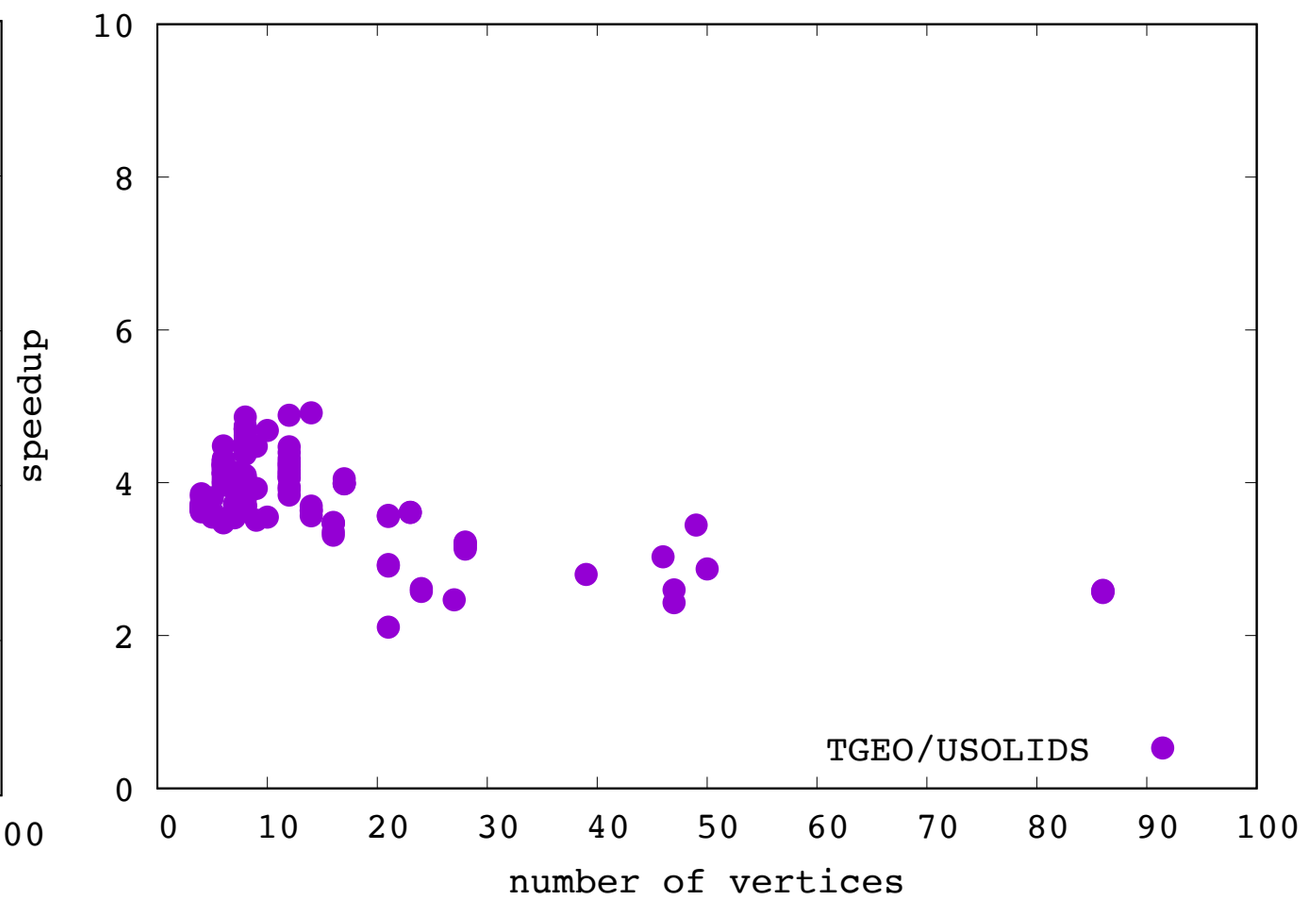
SExtru Performance (2)

Speedup DistanceToOut



mean gain >**5ish** over existing G4/TGeo implementations

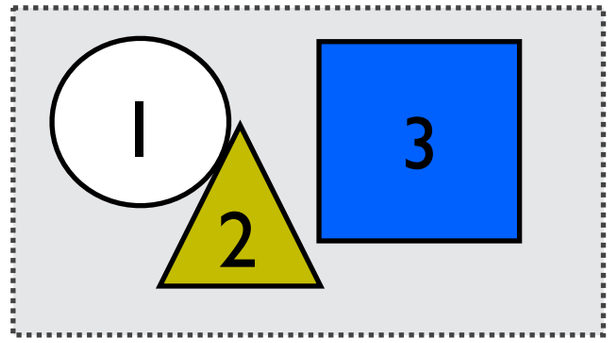
Speedup Contains



2 to 5x faster than existing G4/TGeo implementations

The Assembly

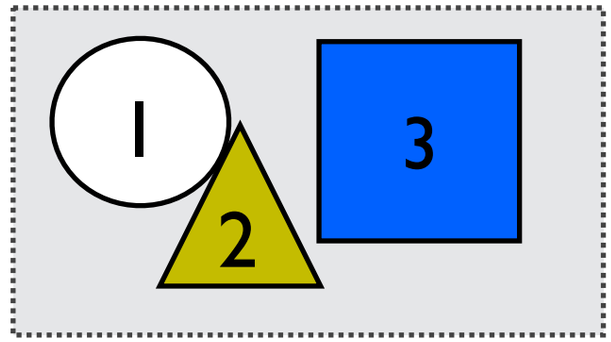
- The **assembly** is a **logical group** of solids
  - used to create repetitions of complex structures
  - but no physical boundary itself
  - no material etc.



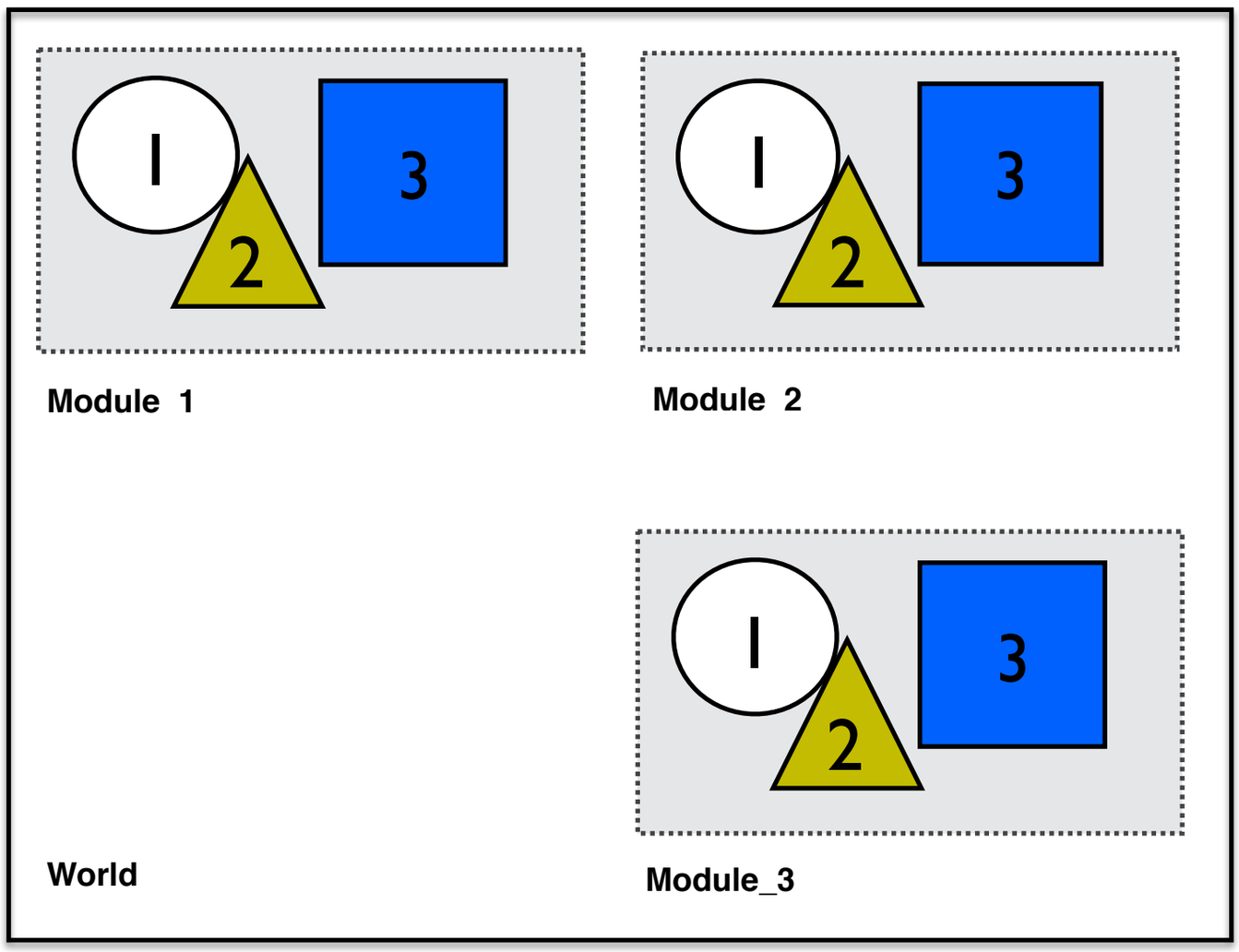
(Assembly) Module

The Assembly

- The **assembly** is a **logical group** of solids
  - used to create repetitions of complex structures
  - but no physical boundary itself
  - no material etc.

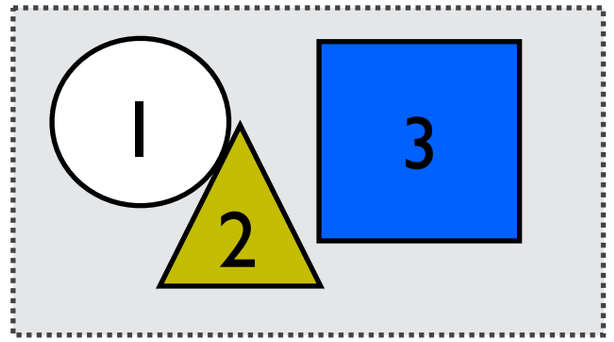


(Assembly) Module

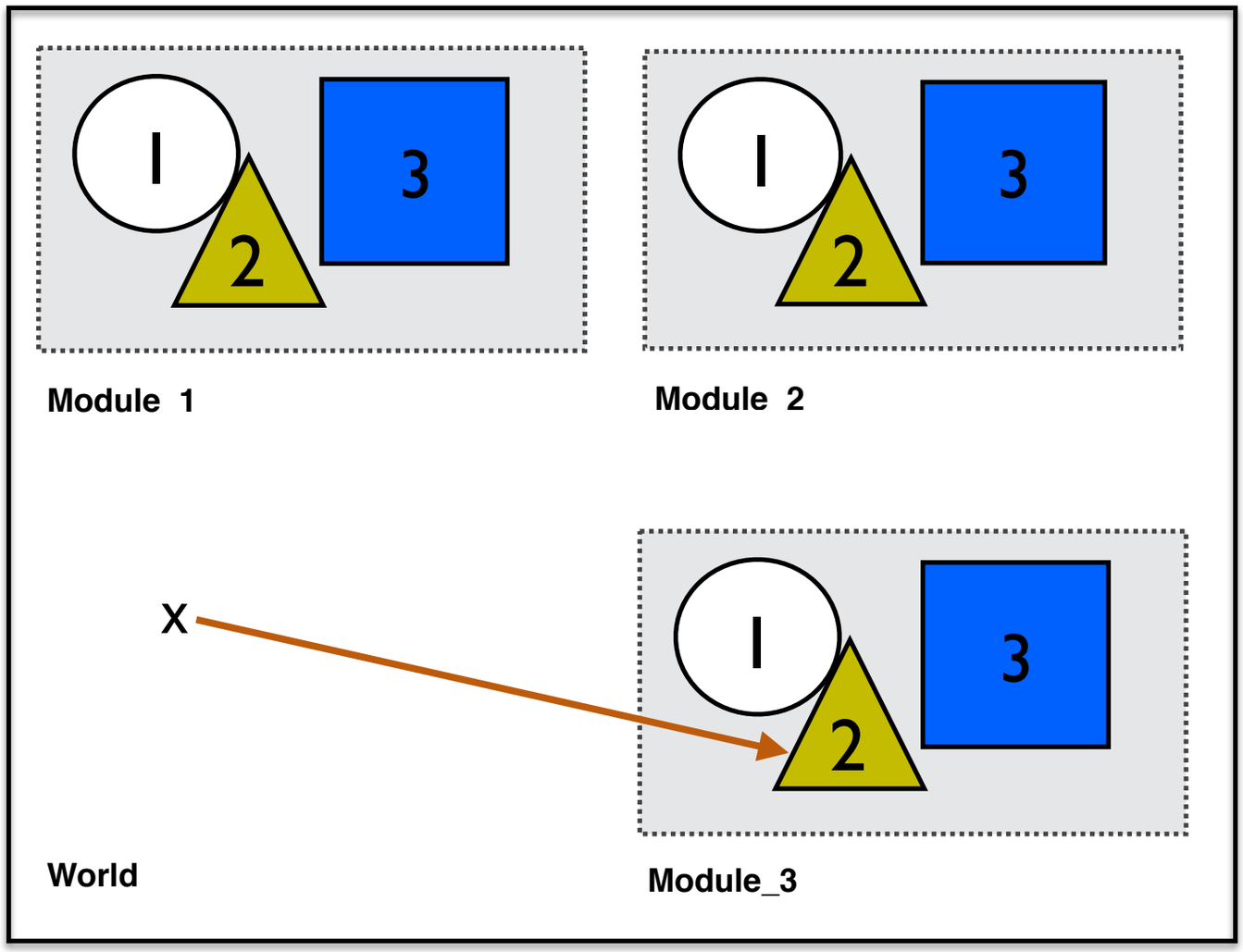


The Assembly

- The **assembly** is a **logical group** of solids
  - used to create repetitions of complex structures
  - but no physical boundary itself
  - no material etc.



(Assembly) Module



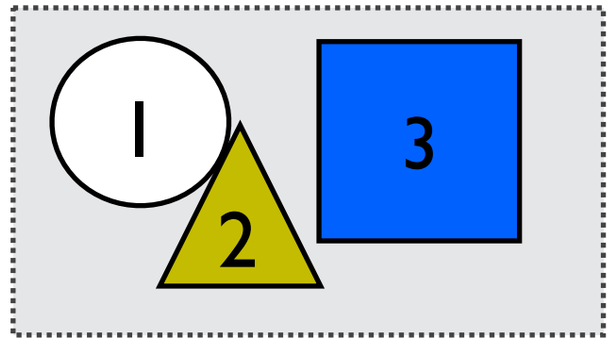
A a track does not stop at assembly boundary but "NavigationHistory" keeps track which assembly placement was entered

/ World

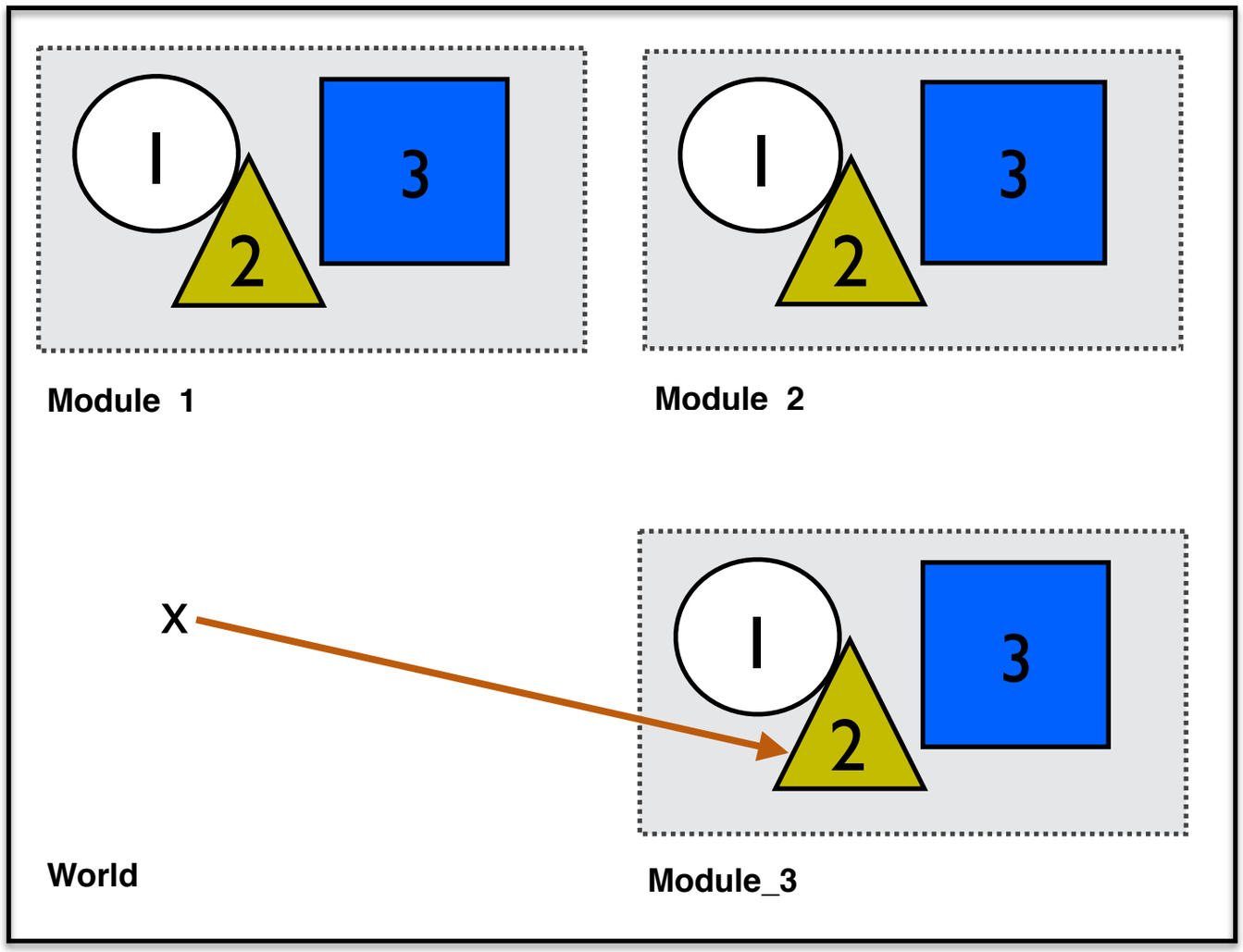
/ World / Module_3 / Solid2

The Assembly

- The **assembly** is a **logical group** of solids
  - used to create repetitions of complex structures
  - but no physical boundary itself
  - no material etc.



(Assembly) Module



A a track does not stop at assembly boundary but "NavigationHistory" keeps track which assembly placement was entered

/ World

assembly implementation touches both "solid" and "navigation" features

/ World / Module_3 / Solid2

The Assembly (2)

- A lot used by ALICE, Panda, CBM, … (typically when geometry coming from CAD)

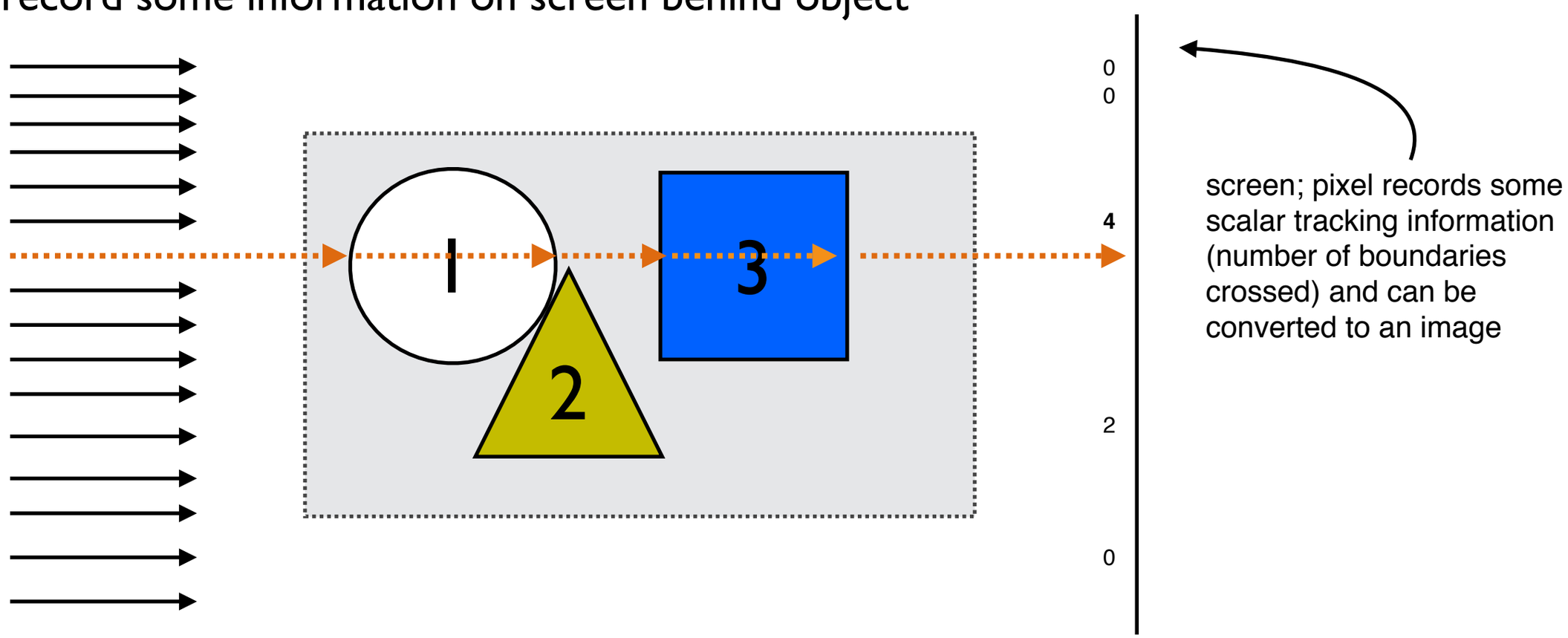
- To use Geant4 on these geometries, need to transforms assemblies into flat list of placements
  - more memory
  - inconvenient for NavigationHistory and scoring (logical- and in-memory representation are different)

- goal was to provide assemblies in VecGeom in the form offered also by TGeo

- implementation is now in place
  - no big magic
  - some general code improvement over TGeo; are now reusing voxel structures and components from navigation classes in favour of less code to maintain

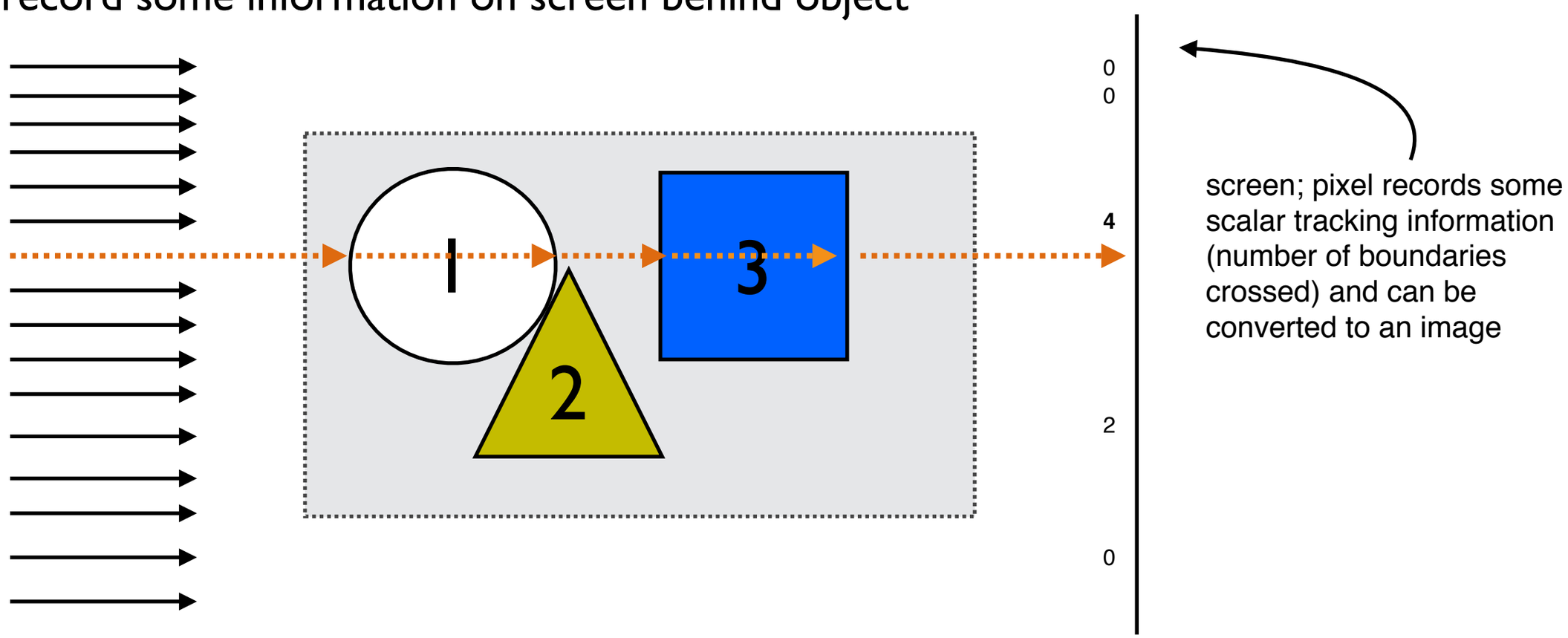
Bringing it all together

- time to test VecGeom (solids + navigation) on complex (ALICE) modules
- one standard test is the "XRayBenchmarker":
  - follow geantinos through geometry - pixel by pixel
  - record some information on screen behind object



Bringing it all together

- time to test VecGeom (solids + navigation) on complex (ALICE) modules
- one standard test is the "XRayBenchmarker":
  - follow geantinos through geometry - pixel by pixel
  - record some information on screen behind object

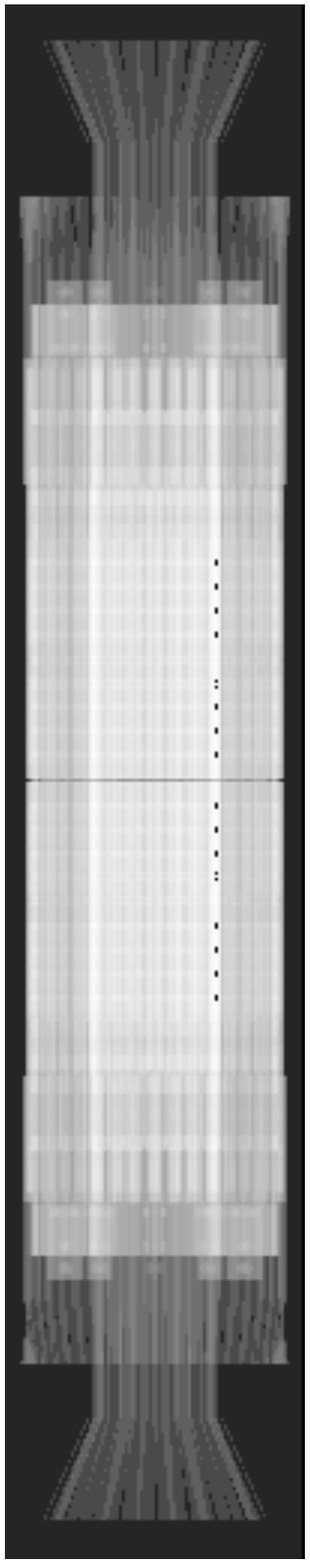


  - can do this also using Geant4 + TGeo thanks to various converters

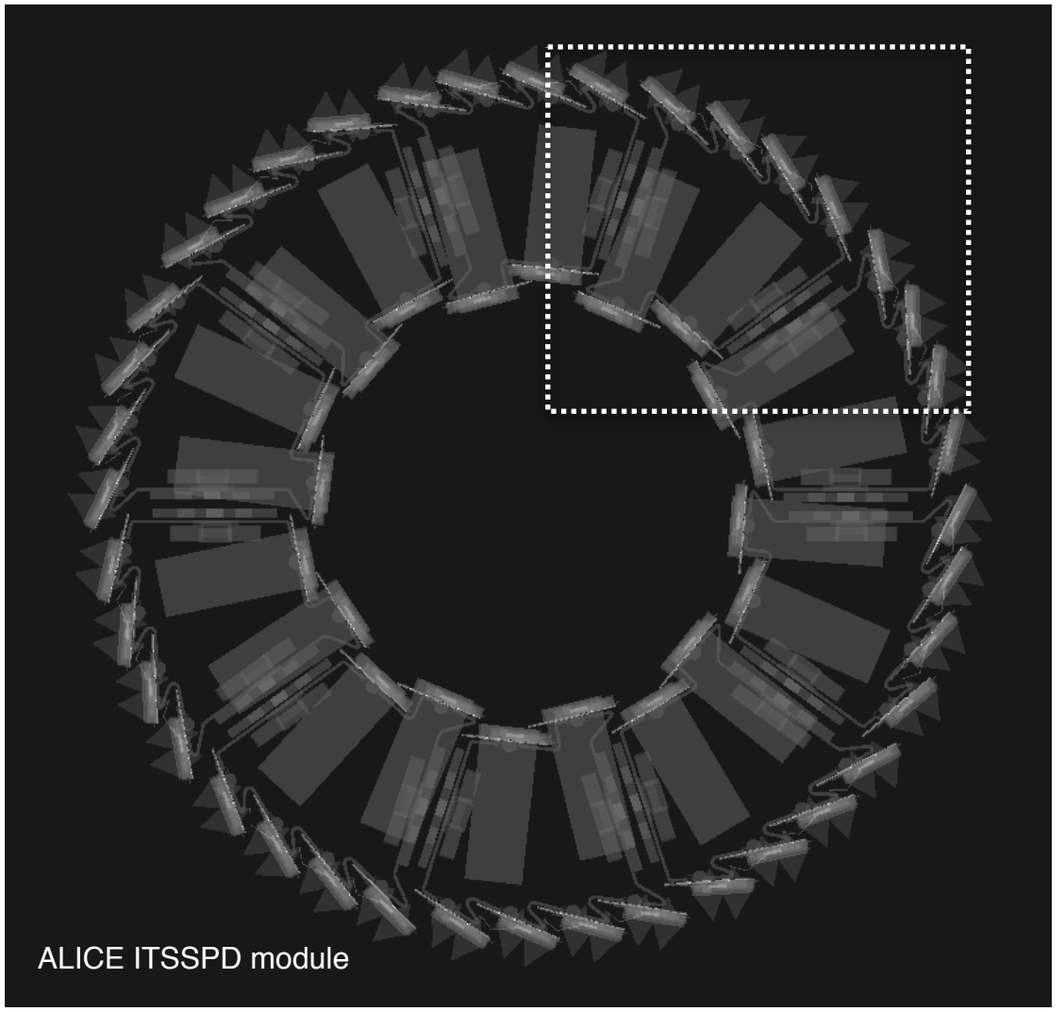
- **perfect** for **validation**; **good** to get a global idea of **library performance**

XRay - Examples and Results

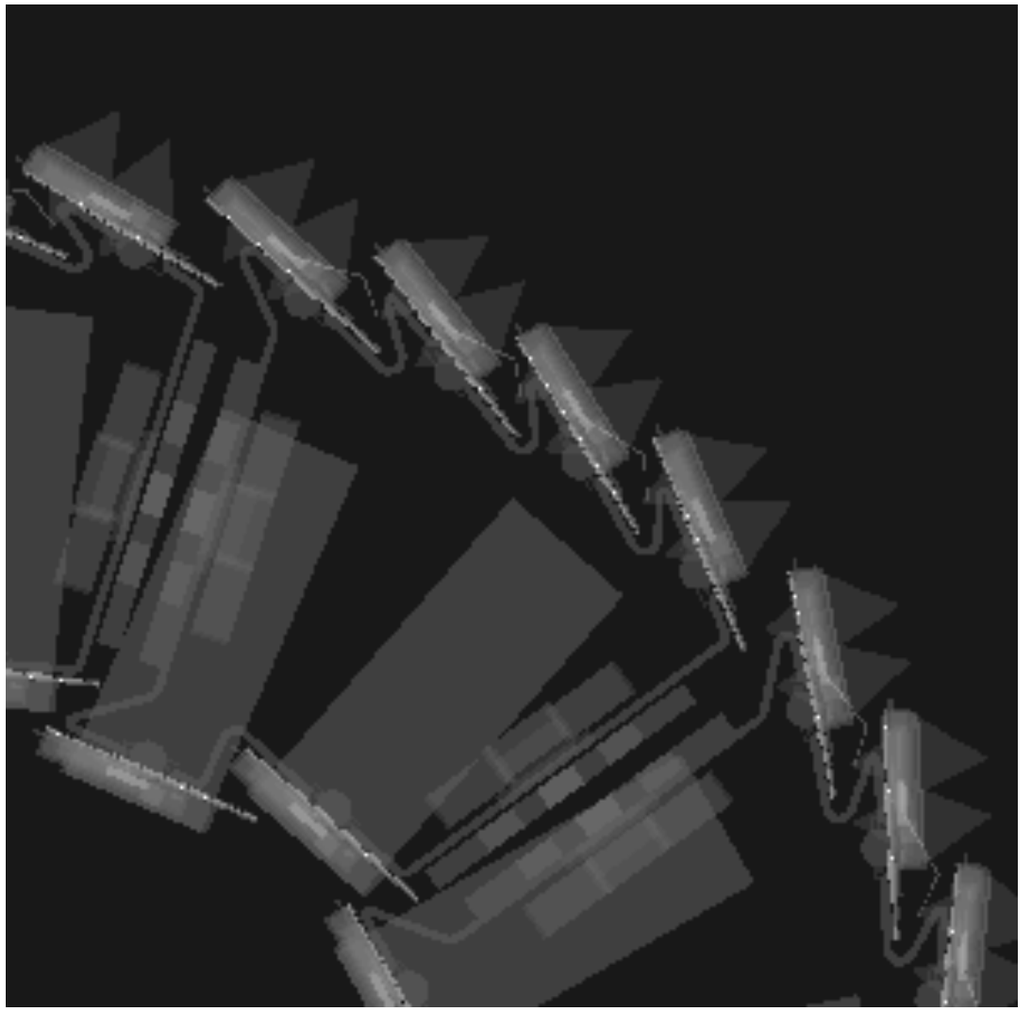
lateral x-ray image



view along z-direction



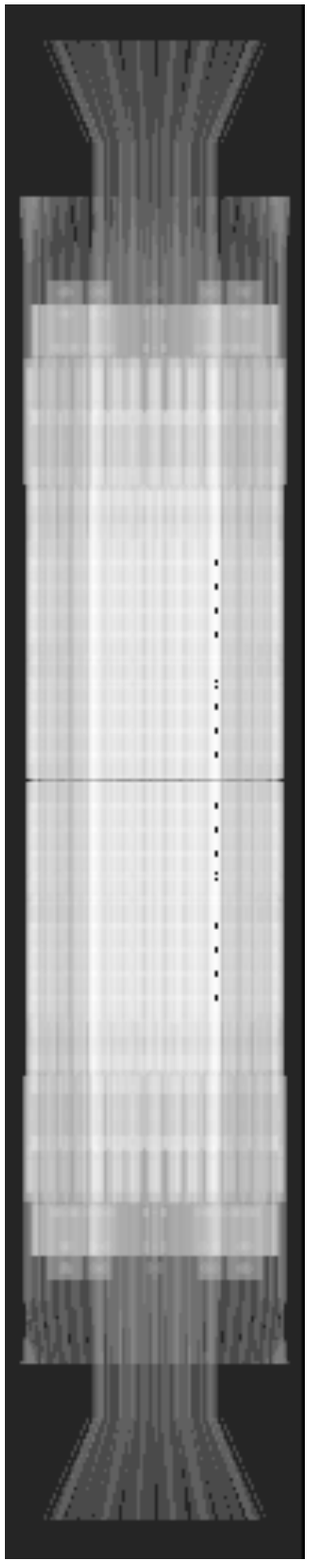
zoom



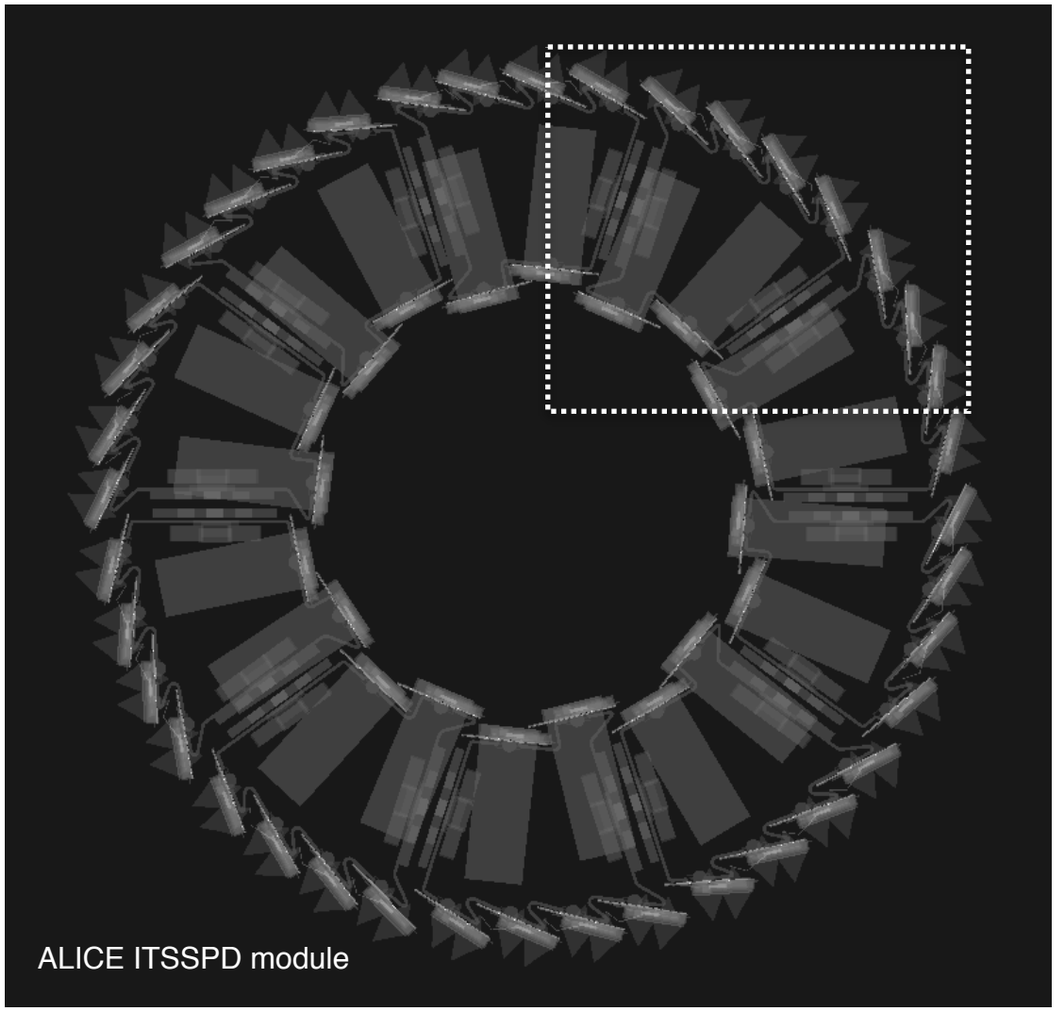
- ITSSPD module containing lots of assemblies, extruded solids, other stuff
- perfect agreement between G4/TGeo/VecGeom

XRay - Examples and Results

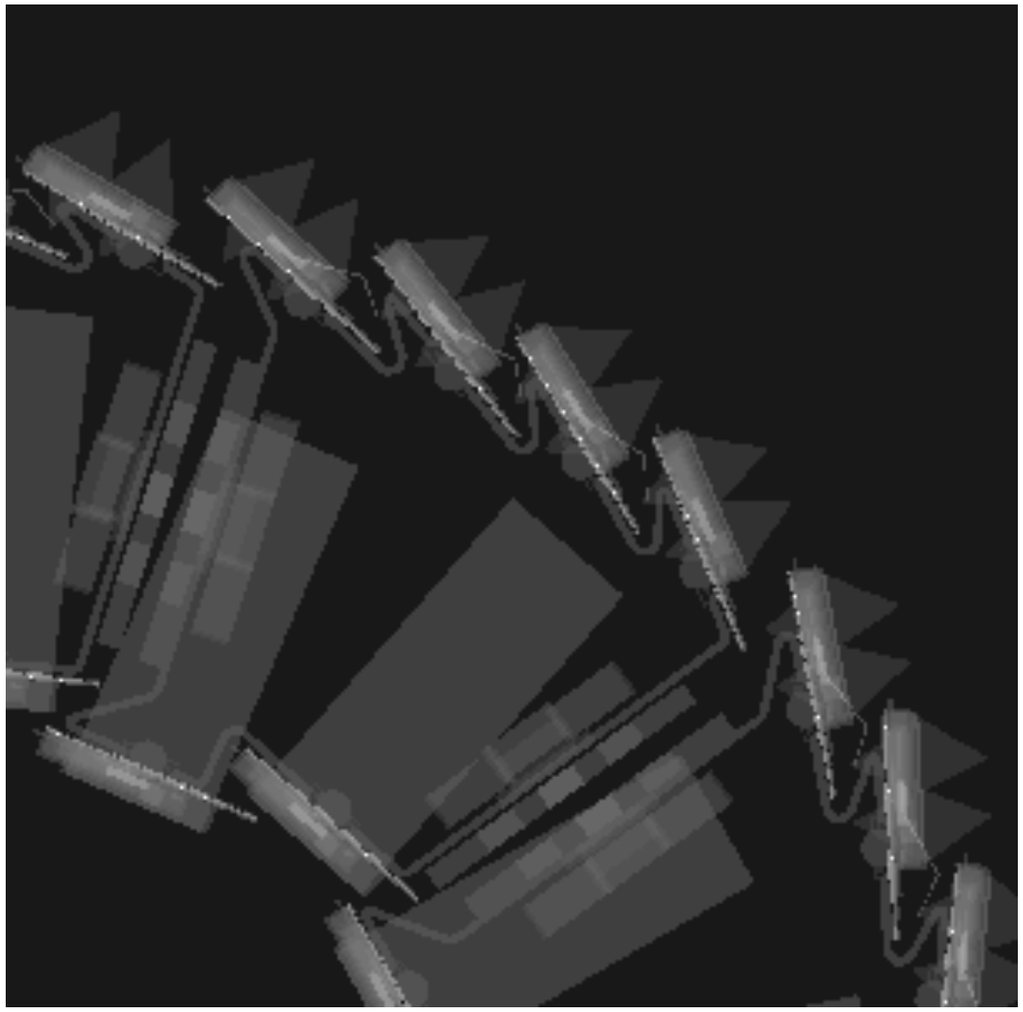
lateral x-ray image



view along z-direction



zoom

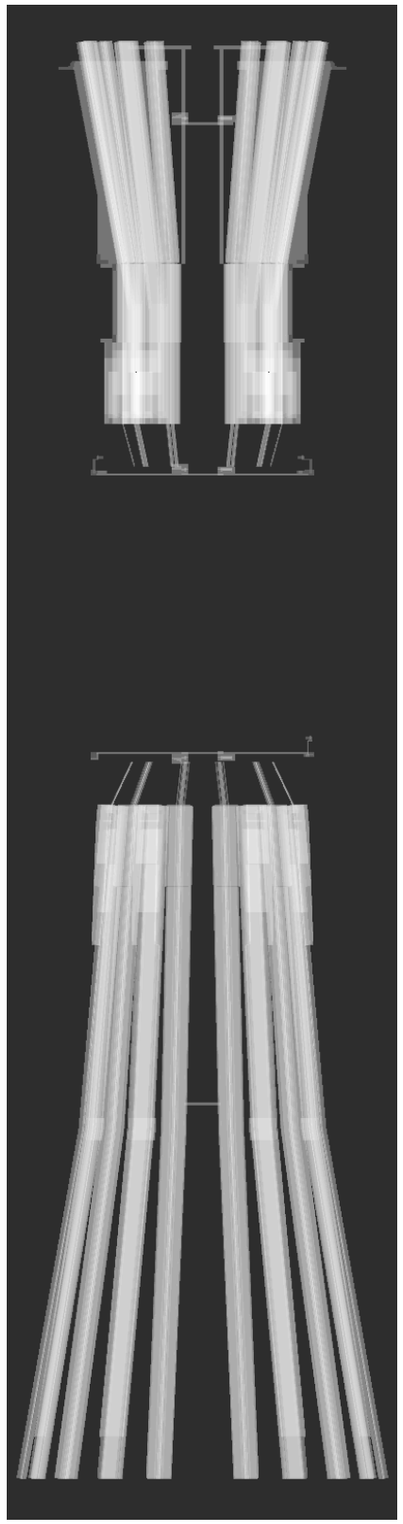


- ITSSPD module containing lots of assemblies, extruded solids, other stuff
- perfect agreement between G4/TGeo/VecGeom
- Overall preliminary CPU improvement for VecGeom (for this example)

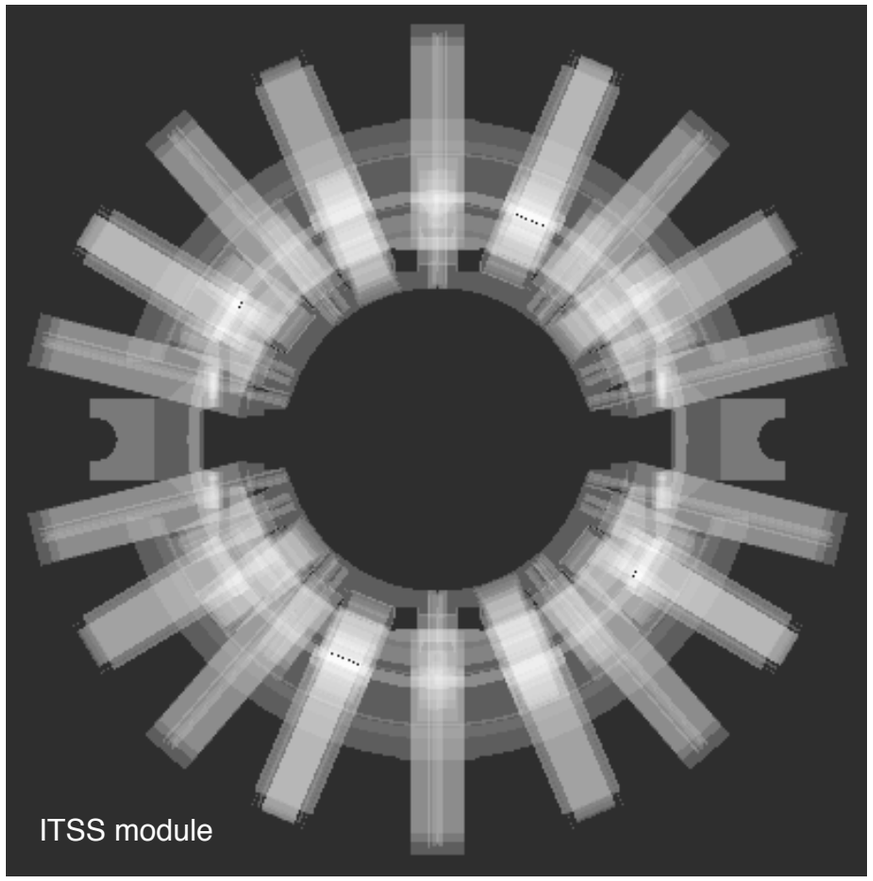
“along z-direction”	2.6x G4	3.3x TGeo
“lateral”	9.1x G4	2.8x TGeo

testing/benchmarking (2)

lateral x-ray image



view along z-direction



“z-direction”	3.9x G4	4.1x TGeo
“lateral”	7.7x G4	4.4x TGeo

- indication of good macroscopic performance of VecGeom at solid + navigation level
- now in the process of systematically validating VecGeom for all modules/ parts of a detector description (from simple to complex)
- this process is done for CMS, ALICE, … and will help integration into G4
  - already found and fixed lots of bugs due to this process
- potentially becoming a stress test which can be run regularly

Summary

Extended USOLIDS/VecGeom to be interesting for ALICE

Now ready to perform integration step into (ALICE) simulation/reconstruction