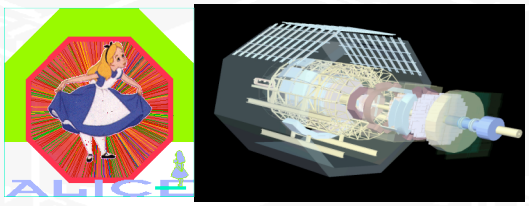


New features in ROOT geometrical modeller for representing non-ideal geometries

René **Brun**, Federico **Carminati**,
Andrei **Gheata**, Mihaela **Gheata**

CHEP'06, 13-17 Feb. 2006

Outline

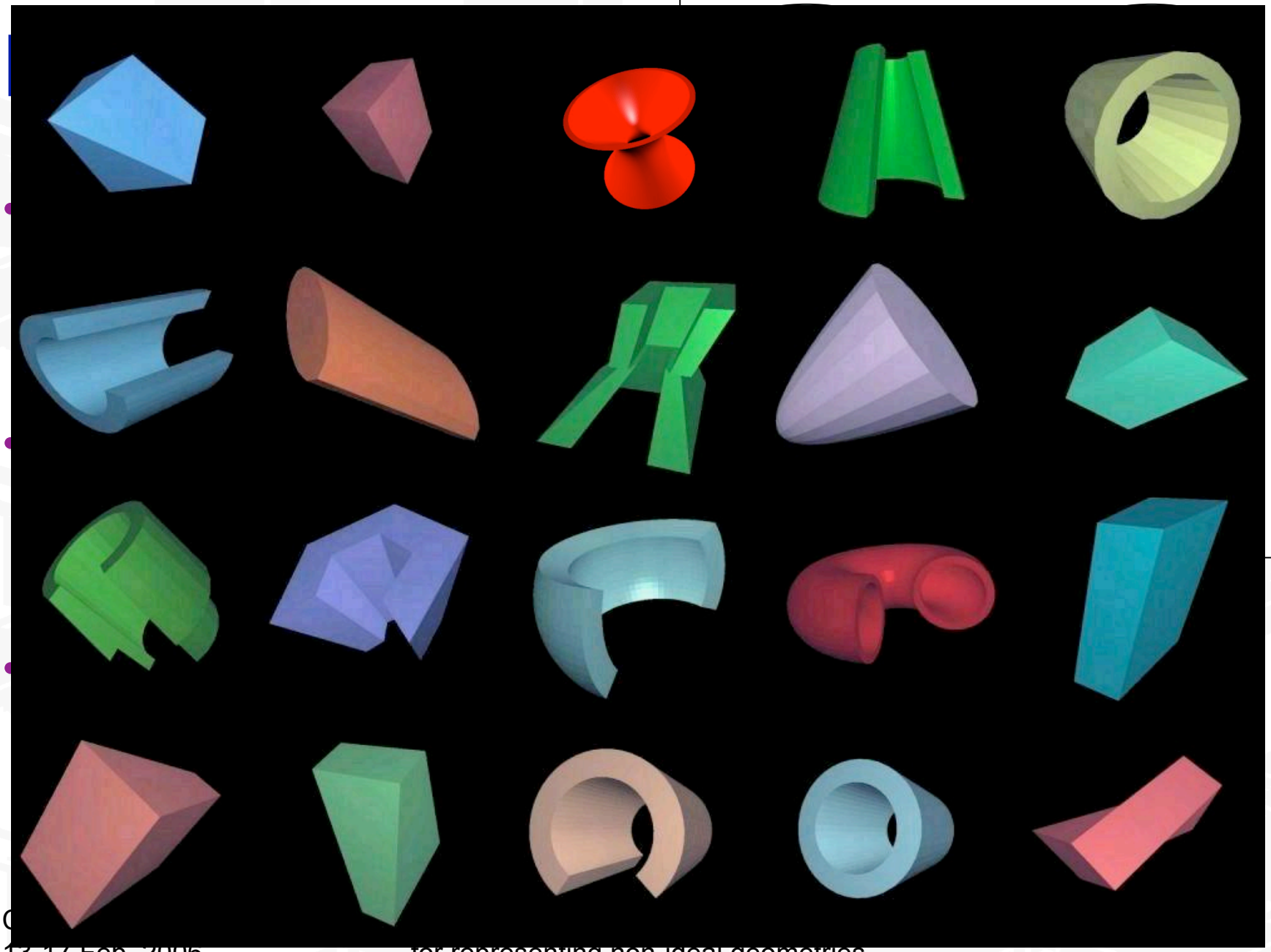
- Ideal vs. real detector
- Misalignment in *TGeo*
- How (not) to use it
- Deformations - scaled shapes
- Assemblies of volumes
- Conclusions

Ideal vs. real detector

- Detectors have generally nice symmetric structures on paper
  - Geometry description and its implementation to specific geometry models used for simulation/reconstruction follows-up
  - In practice, even tracking algorithms and other parts of the code are extensively using assumptions on symmetries
- When building the detector and putting it in place, nothing is as nice as on paper anymore
  - Not taking this into account is non-affordable
    - Degradation of resolution/efficiency (in the best case)
    - No clear recipes to solve the problem (just workarounds)

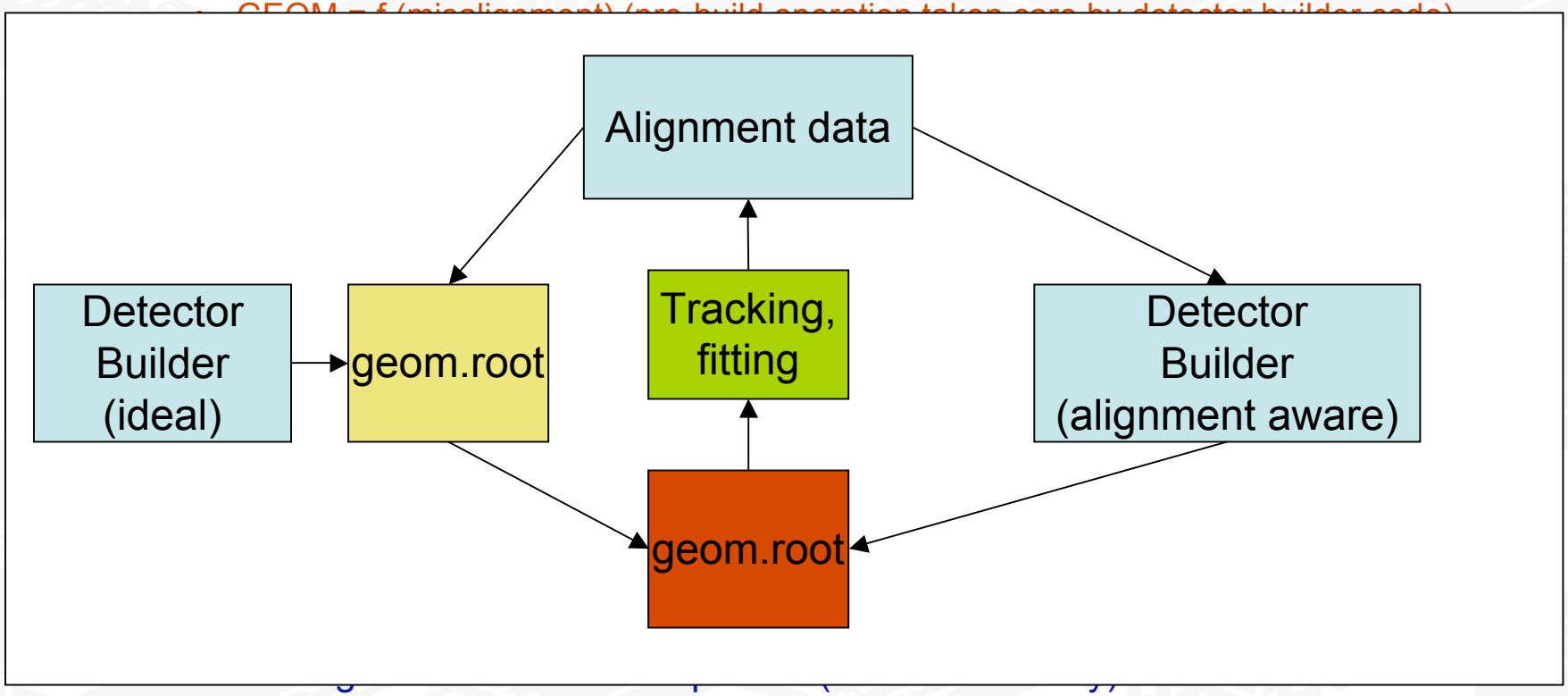
Ideal vs. real detector (2)

- What people usually do:
  - Simplified geometry in the reconstruction framework
  - Start with survey information, then try to “re-align” data in order to minimize track fit residuals, then patch this geometry with the resulting information
- Eventually simulation ≠ reconstruction geometry
  - The difference between the 2 become un-maintainable
  - Simulating the misalignment effects become “bricolage” in the best case – people really learn how to deal with this problem quite long after the first data taking
- Tools to deal with mis-aligned geometries practically not existing
  - Geometry problem => geometry tools



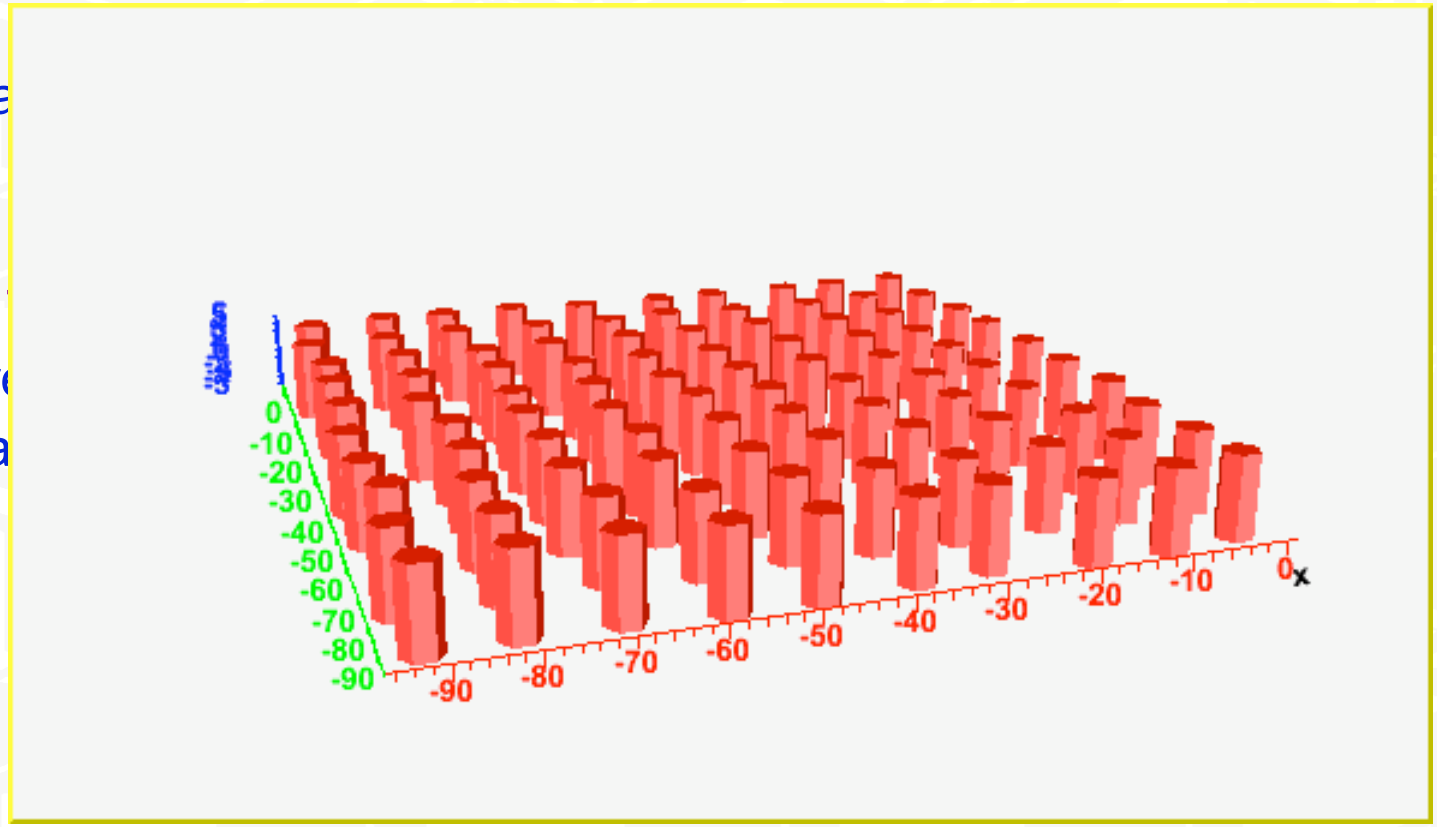
Two approaches

- Misalignment – aware approach
  - Detector geometry aware of the objects that might move



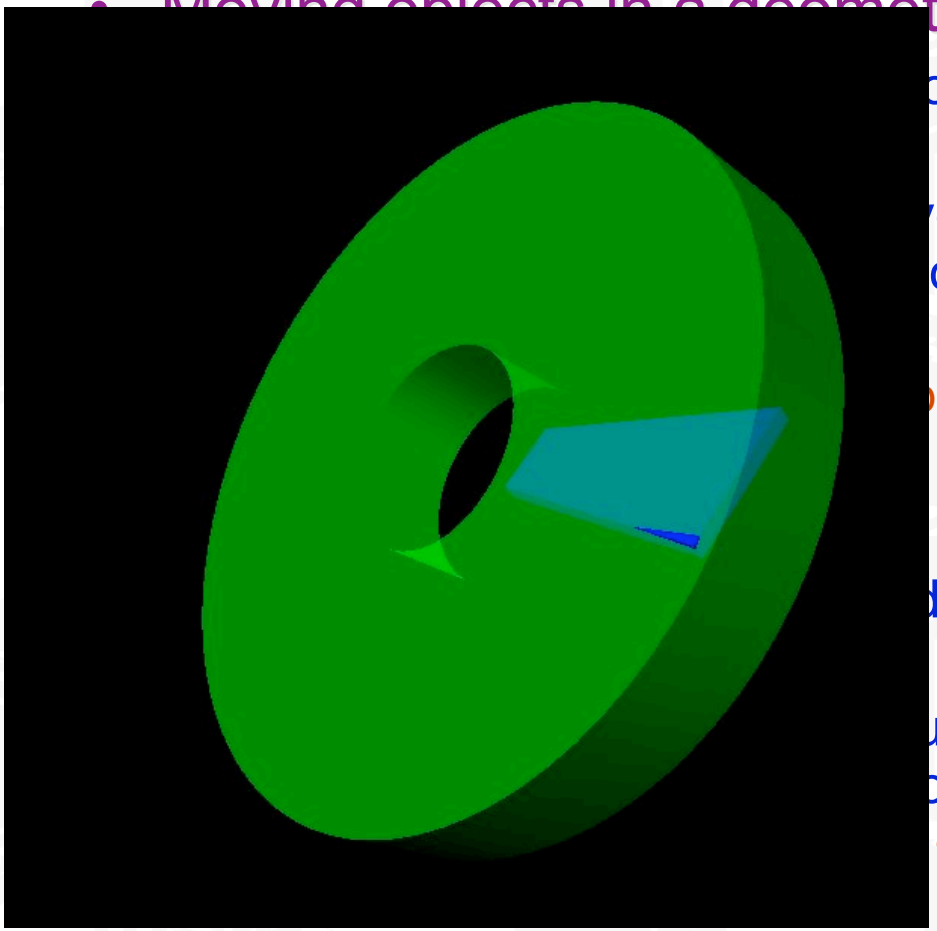
Misalignment “on the fly”

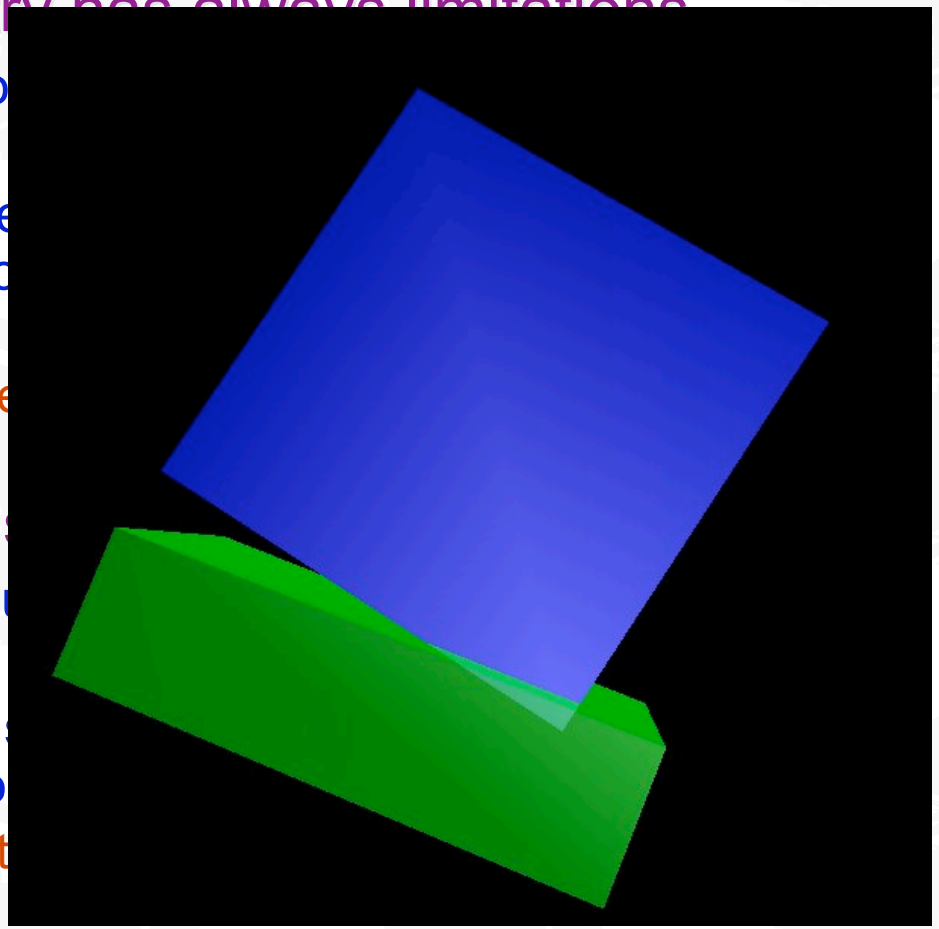
- Describing a single object in the geometry
  - Fully qualified by a path
    -
  - Ma
    -
    -
- How
  - Cr
  - Ca
    -
    -
    -



Caveats

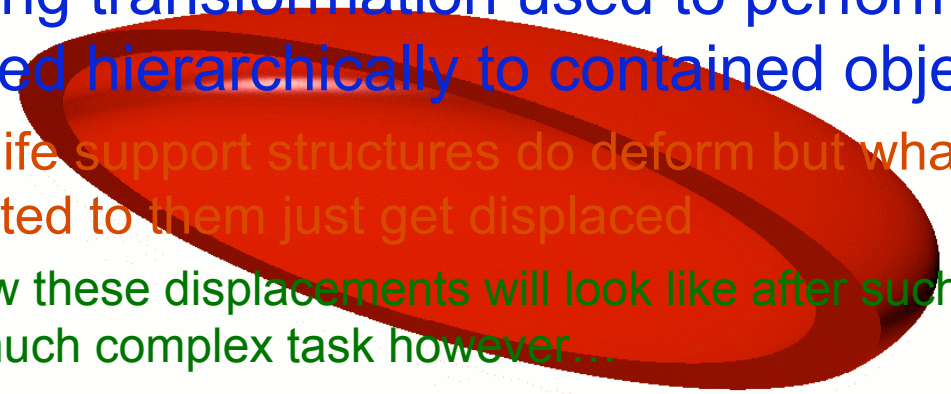
- Moving objects in a geometry has always limitations





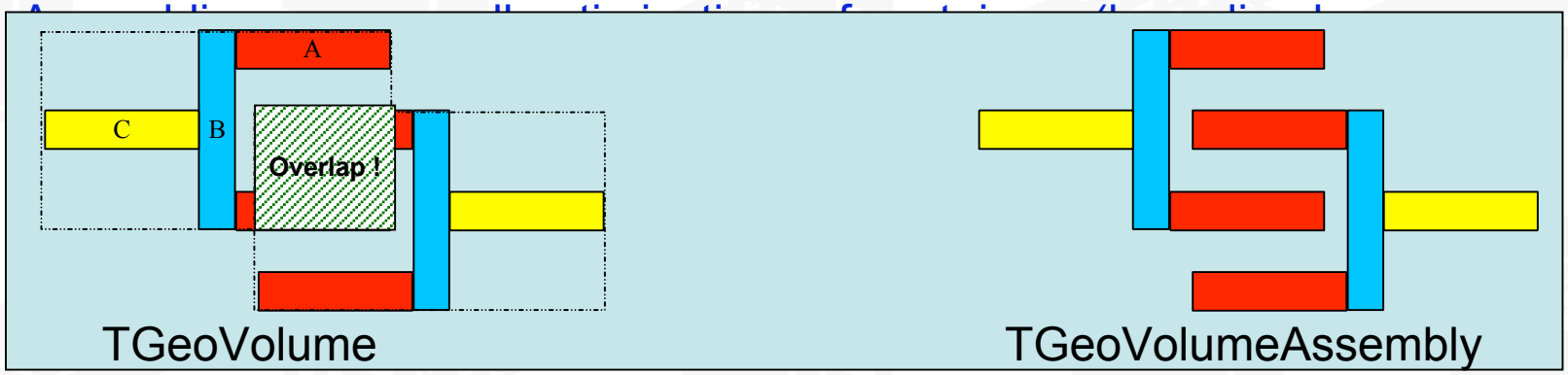
Scaled shapes

- Scaling shapes might be very useful when trying to represent small deformations in the geometry
  - The scaling transformation used to perform this is not propagated hierarchically to contained objects
    - In real life support structures do deform but what is connected to them just get displaced
      - How these displacements will look like after such deformation is a much complex task however…
- One can replace an existing shape with a scaled one during the misalignment with TGeo

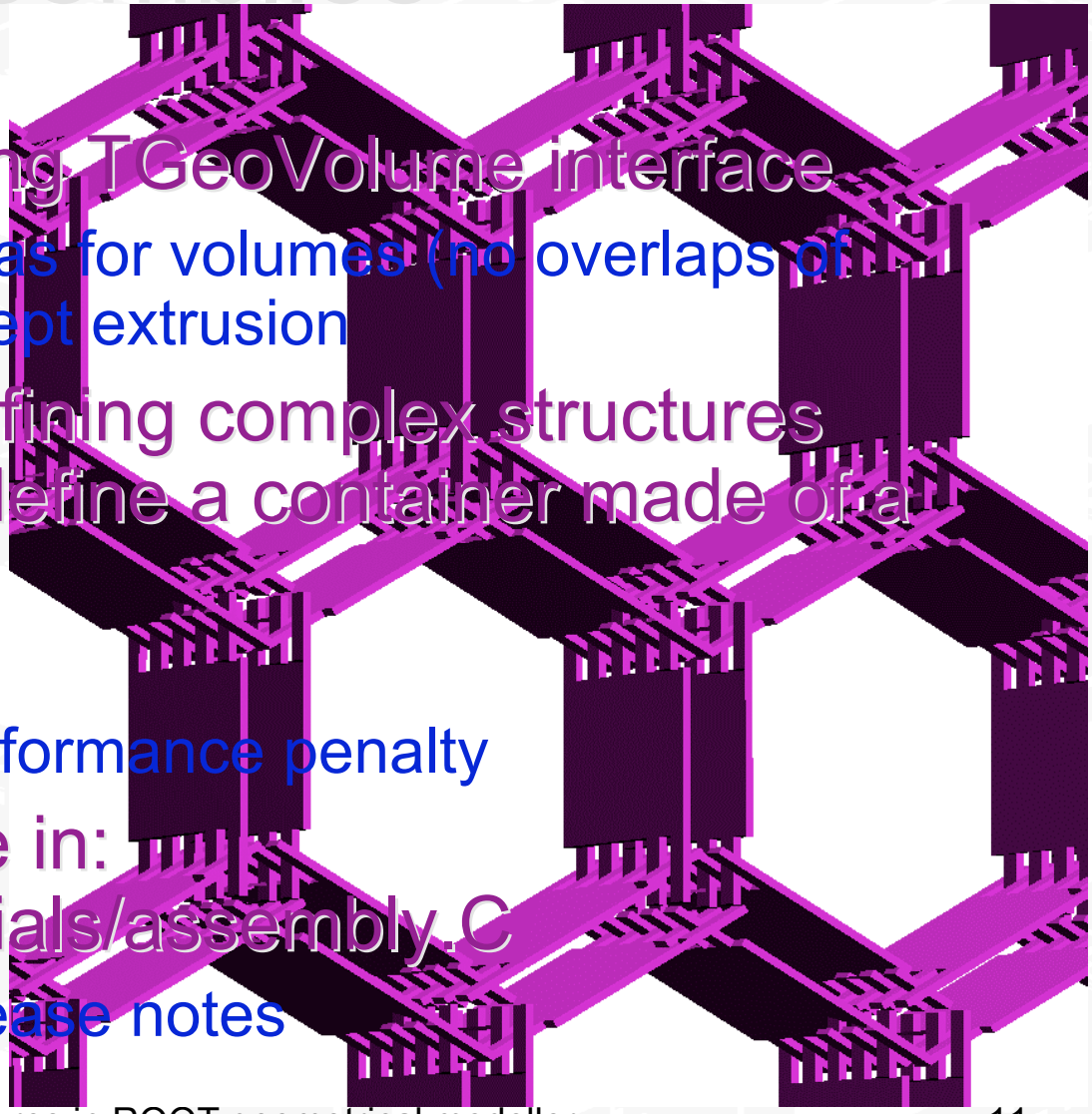


Volume assemblies

- "Software" misalignment not always compatible with *GEANT*-like geometry
  - Containment architecture is the main problem
  - Something needed to allow misalignment but to preserve all benefits of containment (navigation speed)
- Assemblies are defined in *TGeo* as unions of several different volumes positioned with respect to a common local frame
  - An assembly is just a volume having no shape container
    - No medium/material needed: a point INSIDE the assembly is always in one of the components



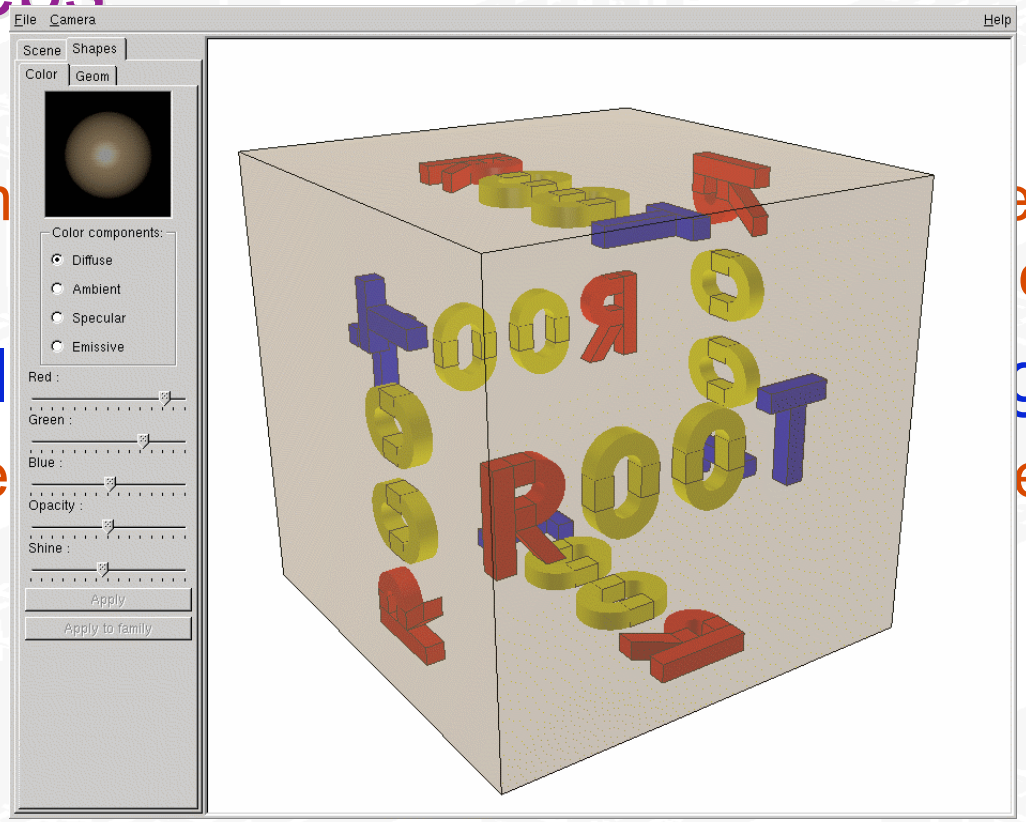
Usage of assemblies



- Easy creation using TGeoVolume interface
  - Same constrains as for volumes (no overlaps of components) except extrusion
- Very useful for defining complex structures where is hard to define a container made of a basic shape
  - Nesting allowed
  - No navigation performance penalty
- Example of usage in: $ROOTSYS/tutorials/assembly.C
  - Description in release notes

Converting existing containers to assemblies

- Two steps:
  - Clone
    - Asm eoVolume container)
  - Actual geometry
    - TGe er, Asm)



Overview

- ROOT geometrical modeller provides support for:
  - Misaligning an existing ideal geometry and using this directly for simulation.
  - Using the same geometry during simulation and reconstruction.
  - Transforming existing geometry containers into assemblies for avoiding possible overlaps during the misalignment procedure.
  - Recursively updating the misaligned objects during the same session when using residuals minimization procedures.
- No changes required in the geometry
  - Still, tracking algorithms should use geometrical information stored in TGeo
  - Misalignment awareness in the way geometry is modeled helps a lot in dealing with the problem
    - Leave some free space for the “movable parts”…
- Examples of usage of these features already exist ROOT tutorials:
  - $ROOTSYS/tutorials/assembly.C
  - $ROOTSYS/tutorials/geodemo.C