Hierarchical Models

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

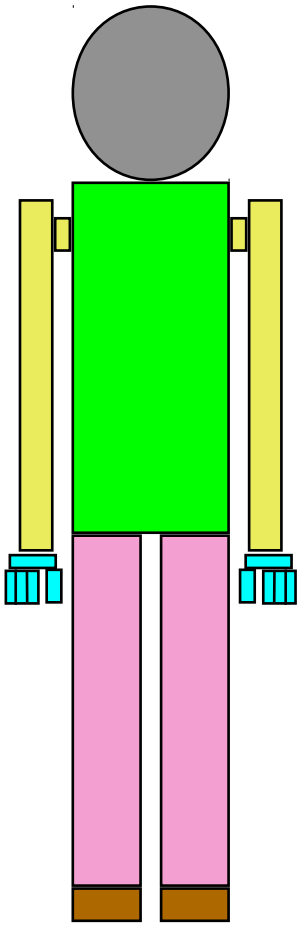
Hierarchies for 3D Modeling

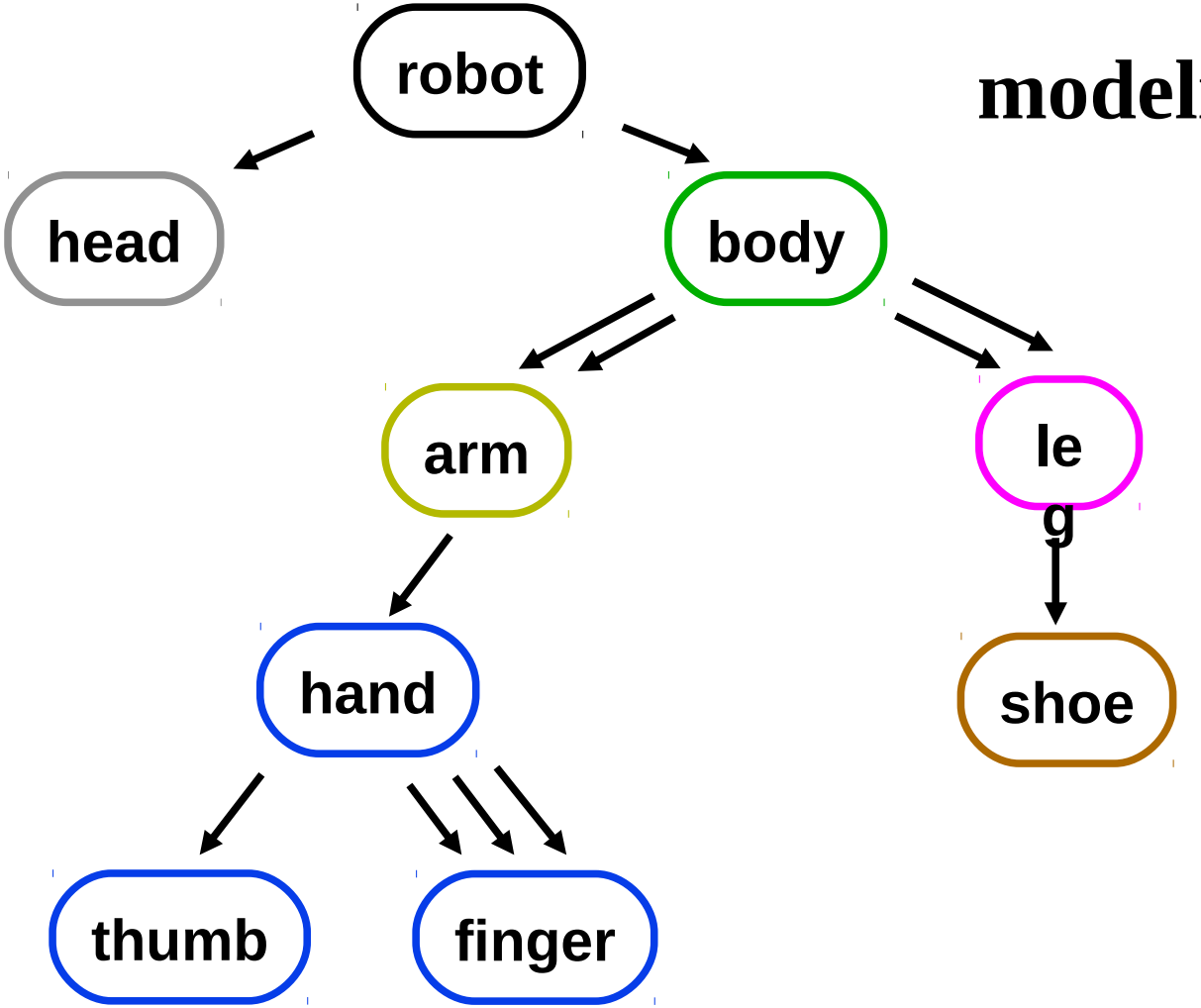
- **Bottom-up modeling**
  - Complex models are assembled from simpler ones
  - During modeling, components are often re-used (building components, standardised parts)
- **Databases of 3D objects**
  - In mechanical and civil engineering, standardised parts are frequently used
- **Parametric models**
  - Individual object instances can be customised

Hierarchical Models

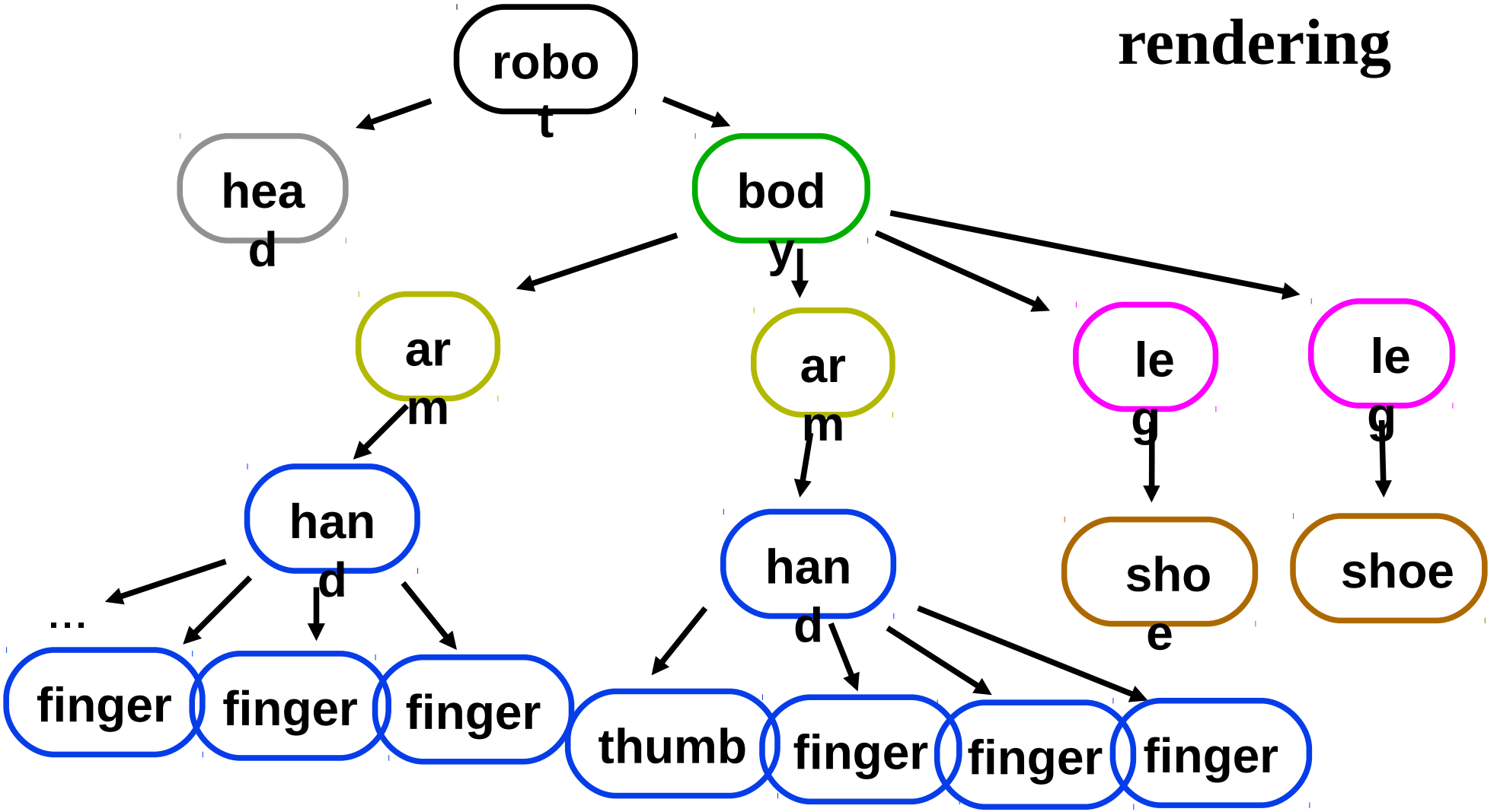
- A scene consists of objects
  - Objects contain of components
    - » Components are made up of parts
      - Parts consist of...
- Hierarchical modeling is simple and effective
  - Entire ready-made model components can be stored in a database, and combined by the designer / user
  - Other features:
    - » Attribute hierarchy (inheritance, parametrisation)
    - » Relative transformation matrices (movement only relative to parent node)

Robot: Hierarchical Model





Robot: Scene Graph



Object Storage in Database

- **Global** (implicit) **attributes and parameters**
  - Colour, material, curve approximations, ...
- **Custom 3D elements**
  - Objects, polygons, surfaces, ... (according to object type)
  - Coordinate system
  - Local values of attributes and parameters
- **References to sub-objects**
  - Transformation matrices (relative!)
  - Parameter and attribute modifications

Models in Memory

- Conversion of DAG to a **tree**
  - node **= object instance**
  - Geometrical data is no longer shared
- Coordinates of **vertices**, **control nodes,** surfaces, ..

  After 3D transformations and projections

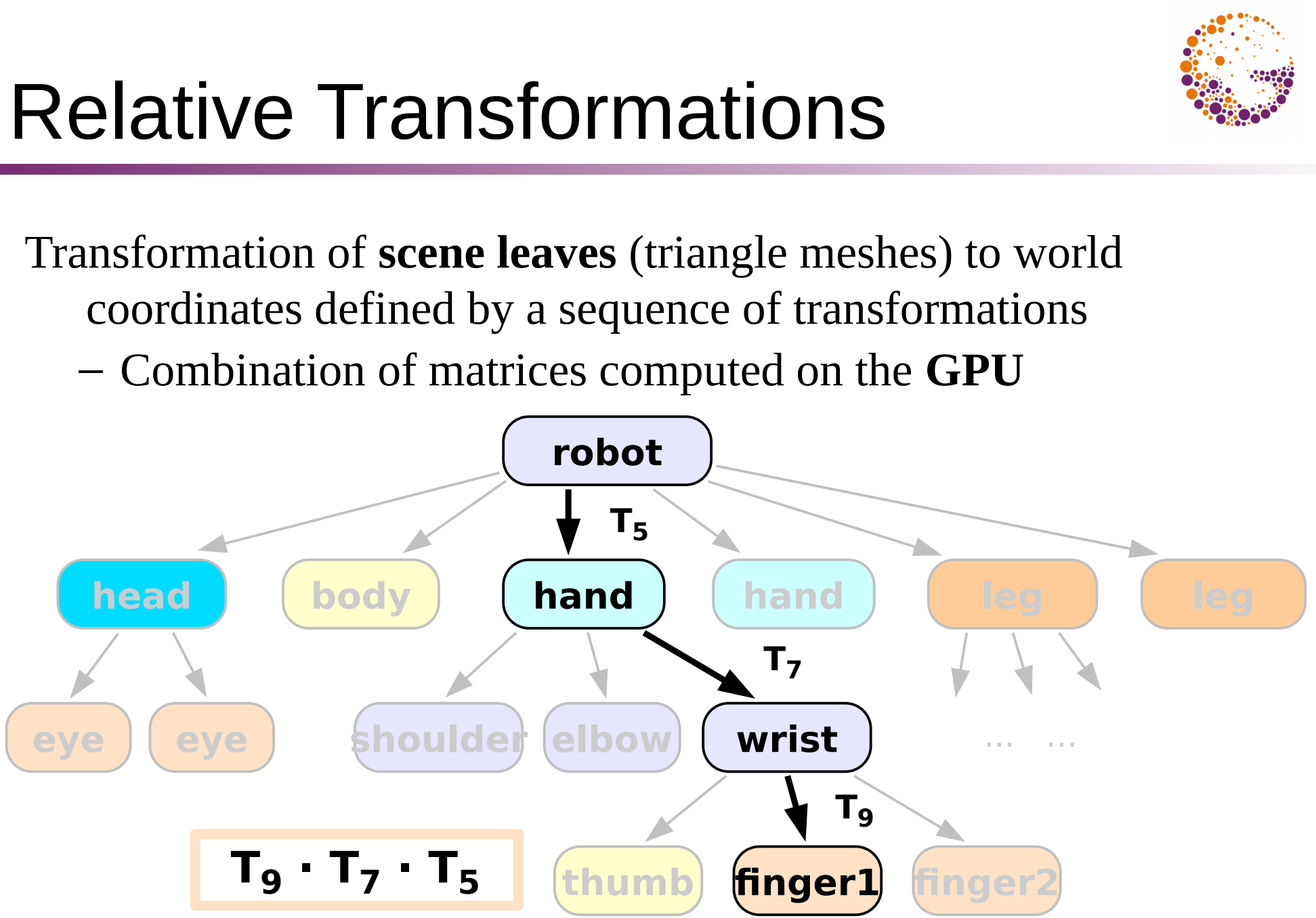
  ~ Relative object coordinates - 3D

  ~ World coordinates - 3D

  ~ Projection coordinates - 2D or 3D (z = depth)

  ~ Display coordinates - 2D (integer)

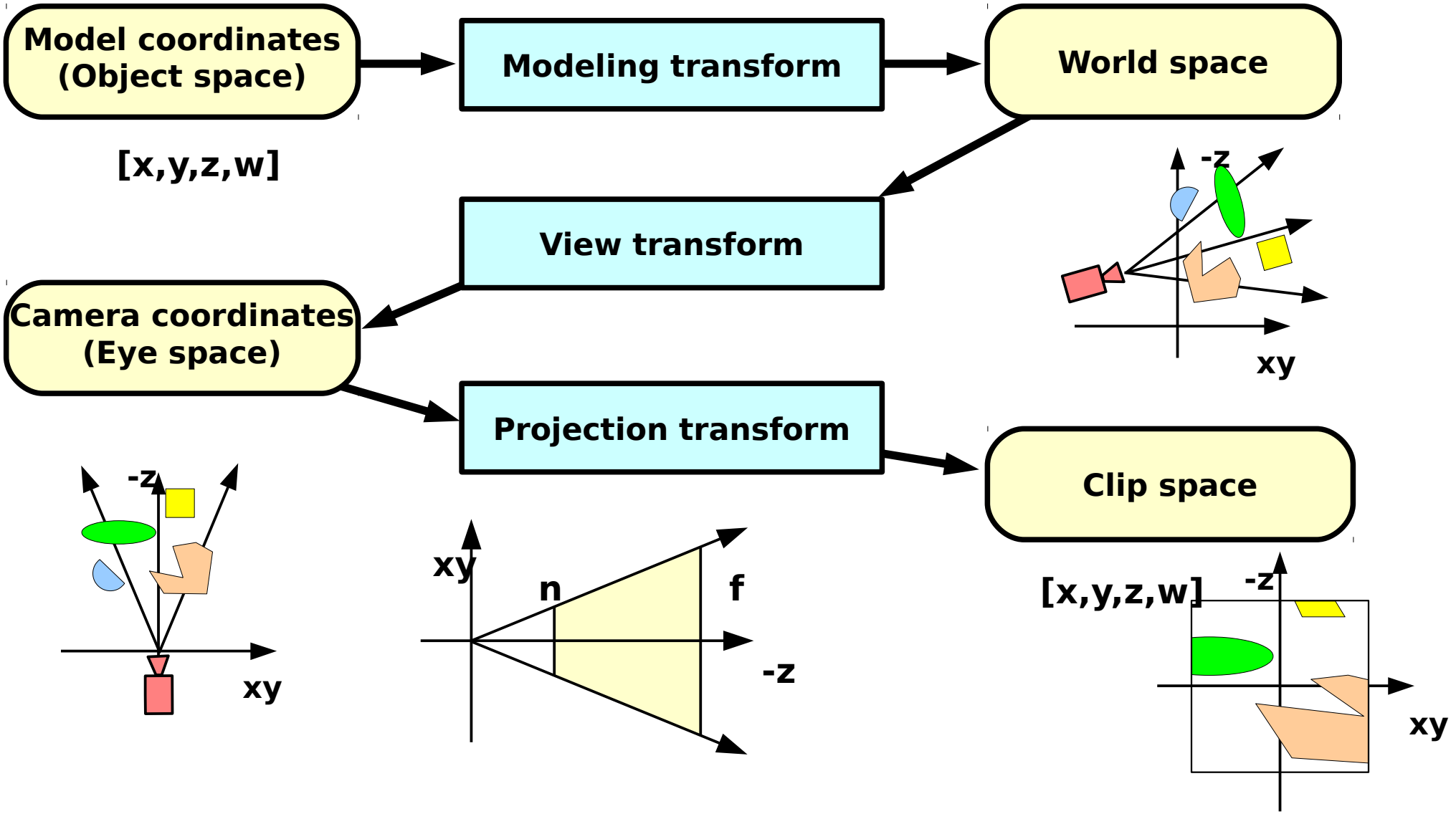
Relative Transformations



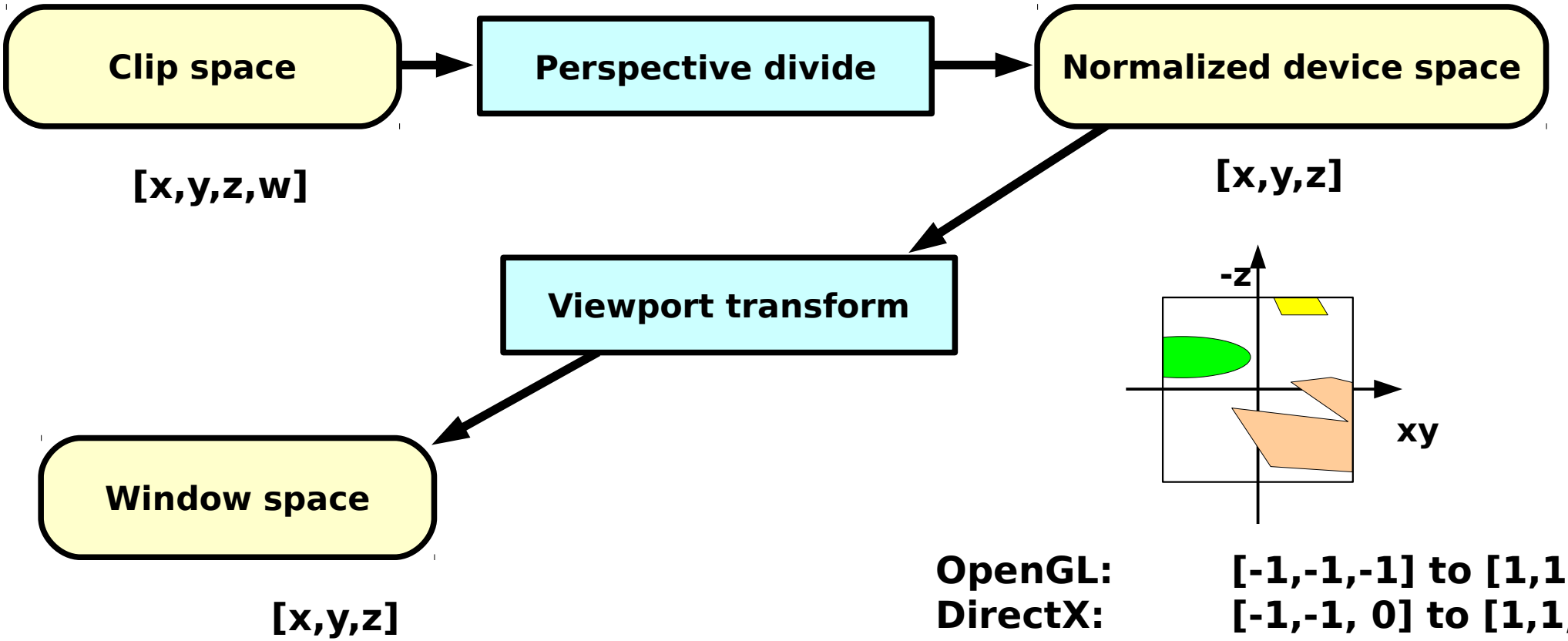
Transformation of **scene leaves** (triangle meshes) to world coordinates defined by a sequence of transformations

– Combination of matrices computed on the **GPU**

Coordinate Systems



Coordinate Systems II

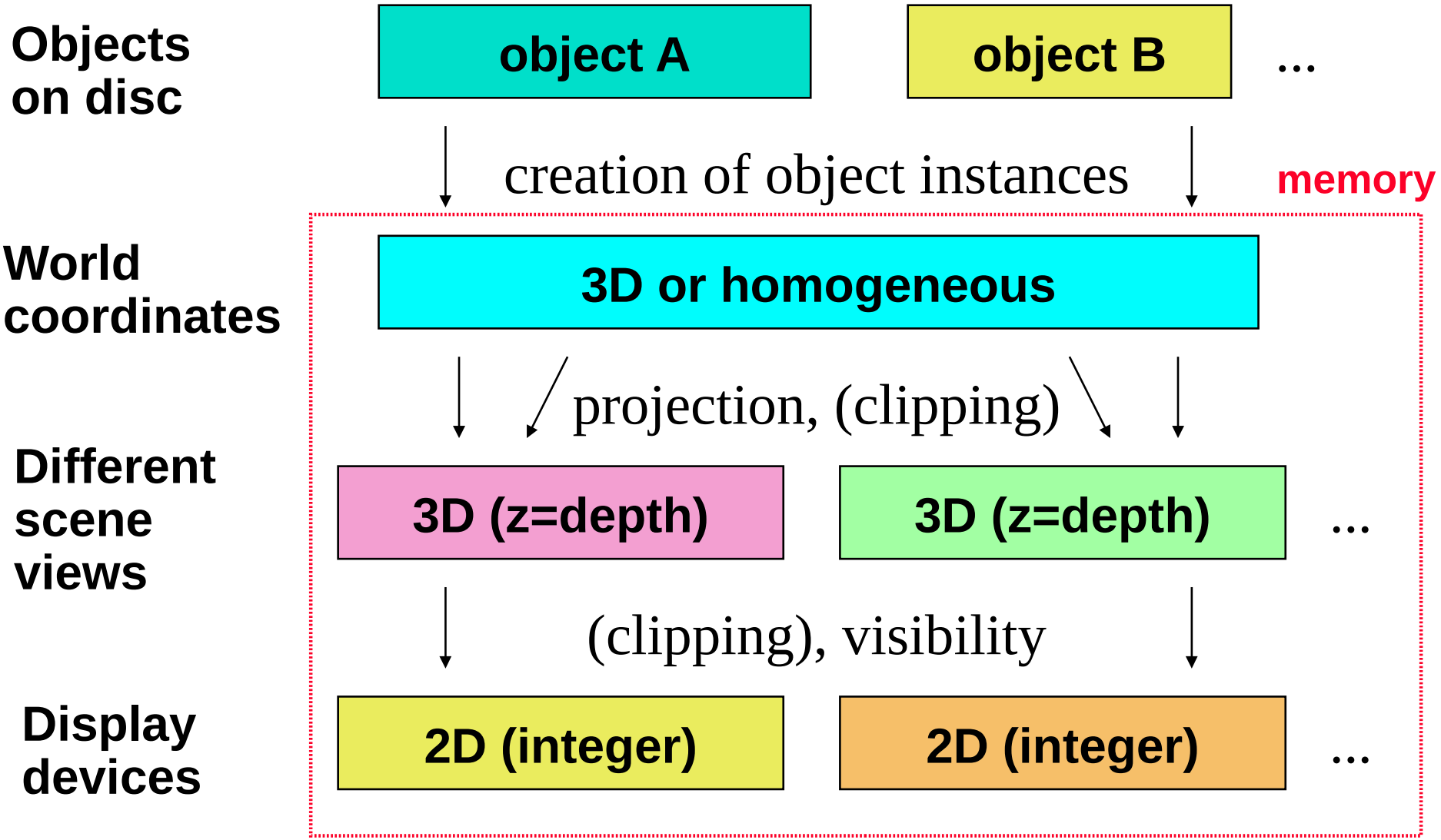


[x,y] actual fragment size on screen
z depth compatible with the z buffer

Coordinate Systems III

- **Model coordinates**
  - Database of objects that comprise the scene
  - Source: 3D modeling applications (3DS, Maya, ..)
- **World coordinates**
  - Absolute coordinates of the 3D world
  - The relative coordinates of object instances are given there
- **Camera coordinates**
  - 3D world → relative camera coordinates
  - Projection center: **origin**, view direction: **-z** (or **z**)

Coordinates (vertices, nodes, ..)



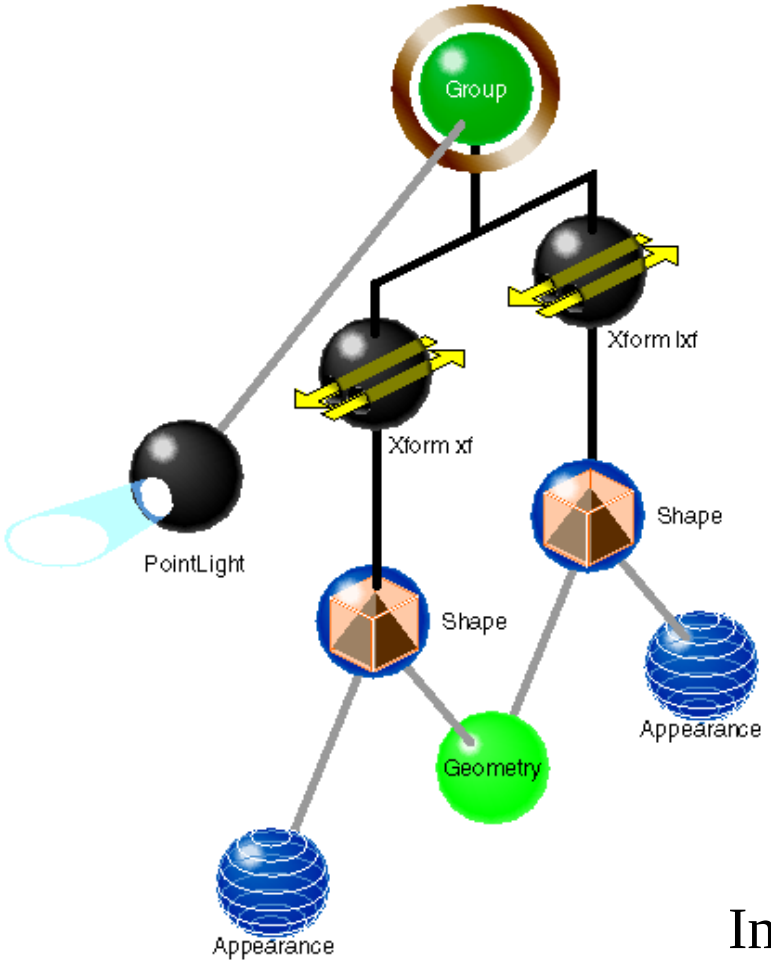
Hierarchical 3D Formats

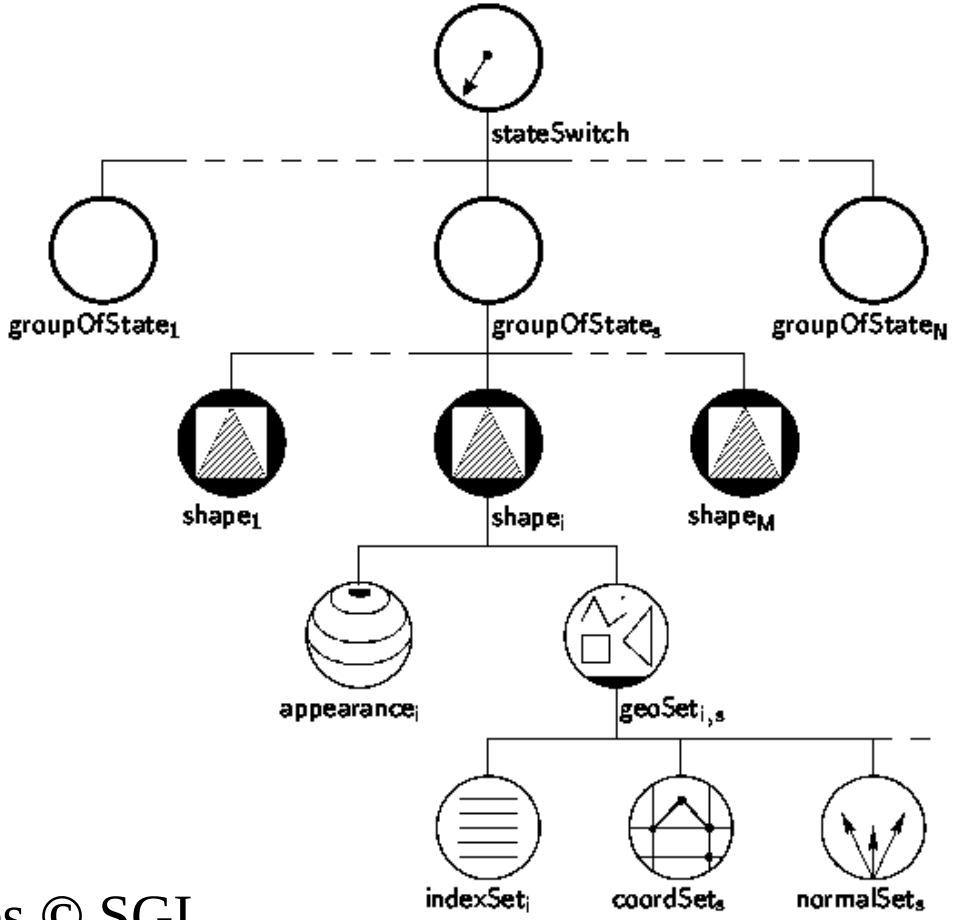
- **PHIGS(+)** (ANSI, ISO)
  - „Programmer's Hierarchical Interactive Graphics System"
- **OpenInventor**, **Performer** (both SGI)
  - Object-oriented OpenGL front-ends
- **VRML** („Virtual Reality Modeling Language")
  - WebSpace (World-Wide Web)
- **OpenSG**, **X3D**, …
- **Input formats** of rendering programs
  - PoV Ray, RayShade, Radiance, …

Scene Graph

- Scenes are represented by a tree (or DAG)
  - *Internal nodes* – transformations, attribute changes, „groups”, selection, … time dependencies
  - *leaves* – geometry (vertices, normals), lights, materials, …
  - DAG – some leaves or geometry can be shared (e.g. common geometry)
- Result is defined by an in-order graph traversal
  - *Internal nodes* modify parameters & state
  - *Leaves* contribute to the actual result (scene primitives)

Scenegraph





Images © SGI

End

Further information:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 285-346