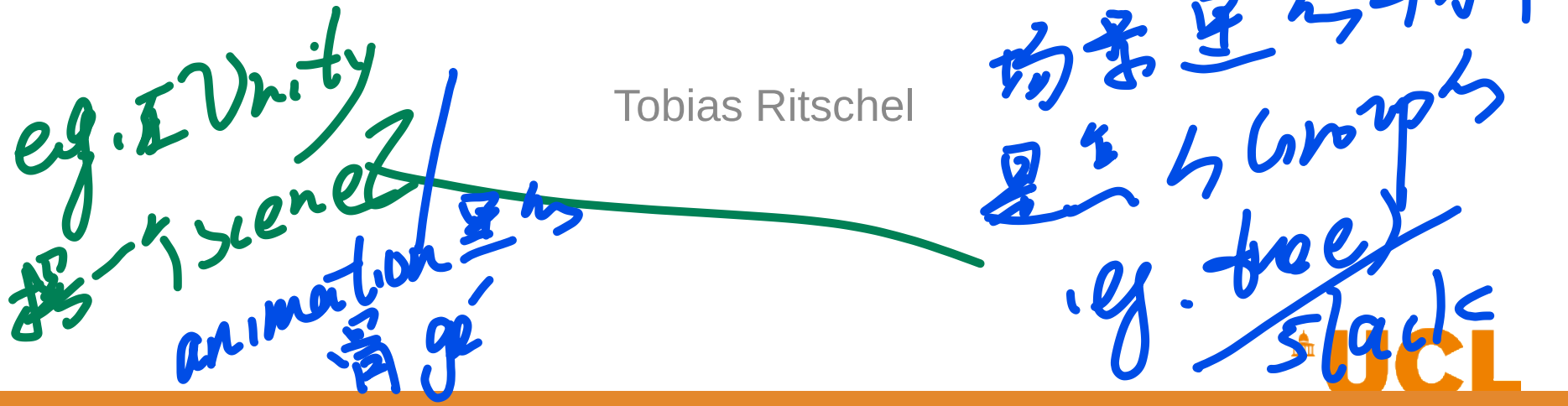
Computer Graphics (COMP0027) 2022/23

Scene Graphs

Tobias Ritschel

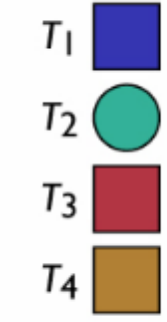


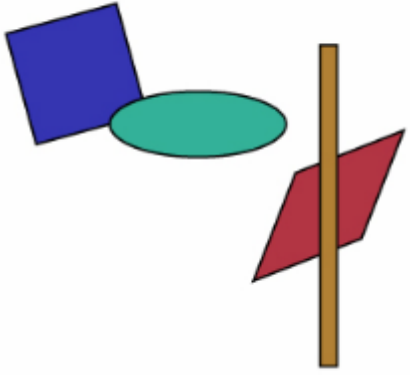
Overview

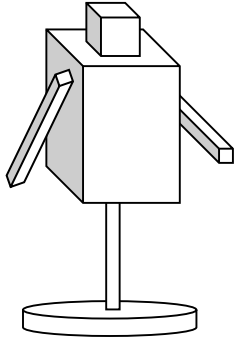
- From a single **object** to a **scene**
- Scene graph
- Local coordinates (LC) vs World coordinates
- Local transformation matrix (LCM) and current transformation matrix (CTM)

Representing Scenes

- List of objects
- Transform for each object
  - can use minimal primitives: ellipse is transformed circle
  - transform applies to points/vertices of object

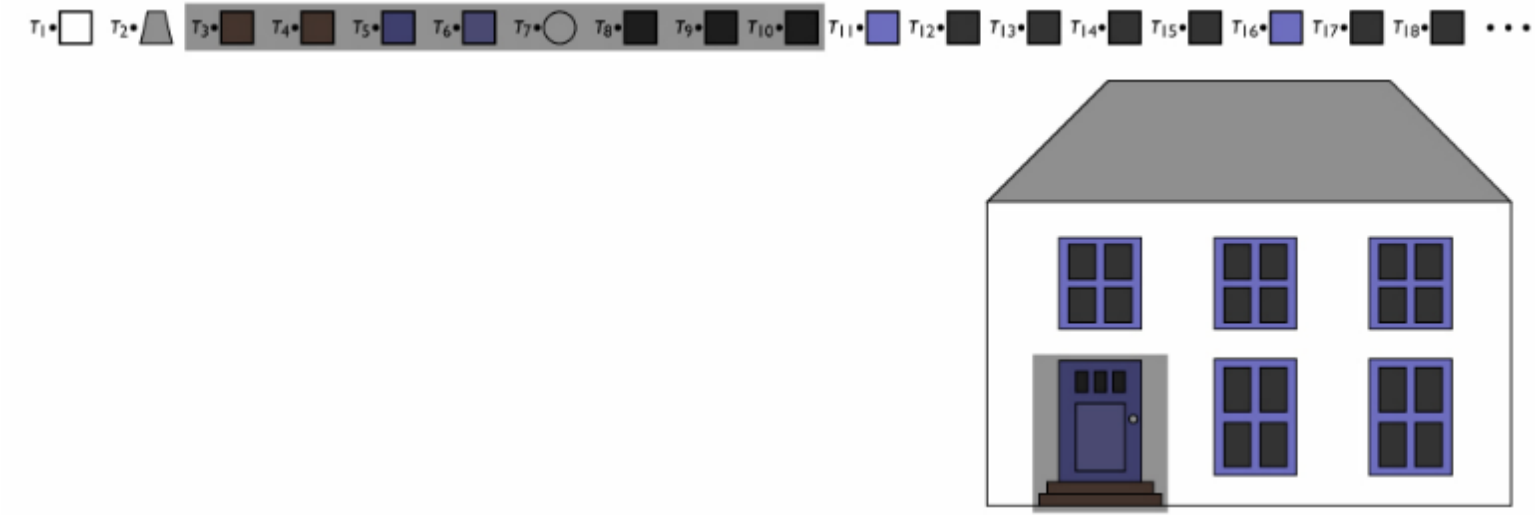






Representing Scenes

- Can represent drawing with flat list
  - but editing operations require updating many transforms

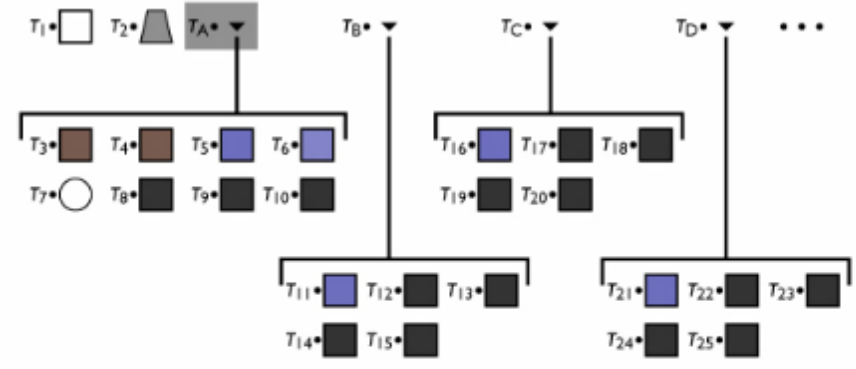


Groups of Objects

- Treat a set of objects as one
- Introduce new object type: group
  - contains list of references to member objects
- This makes the model into a tree
  - interior nodes = groups
  - leaf nodes = objects
  - edges = membership of object in group (and possibly transformations too)

Groups of Objects

- Add group as a new object type
  - lets the data structure reflect the drawing structure
  - enables high-level editing by changing just one node

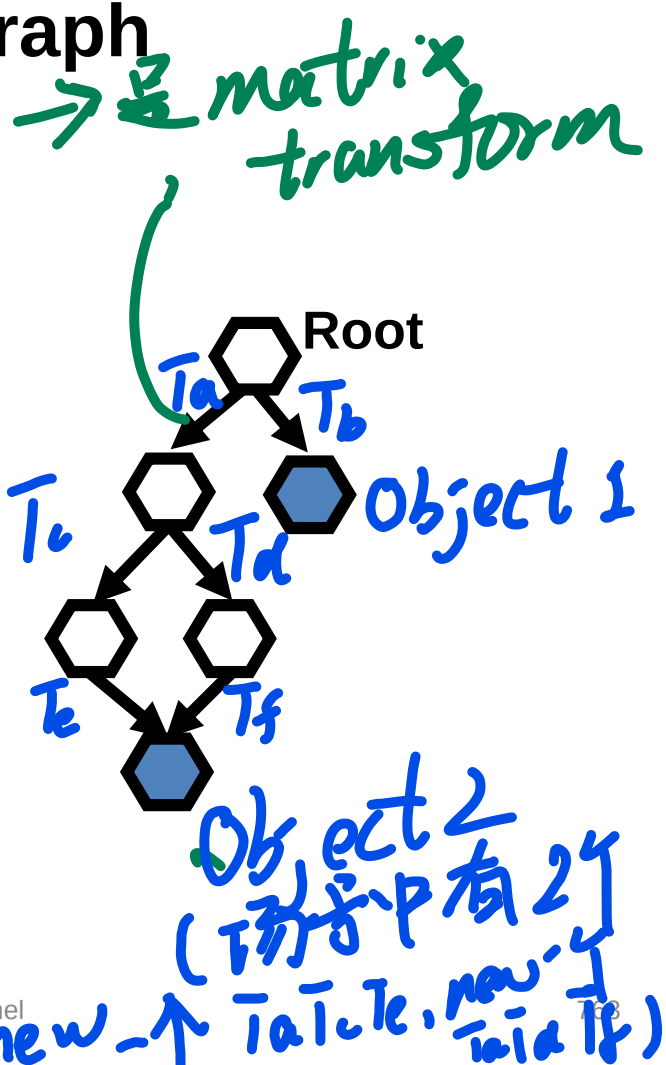


Scene Graph

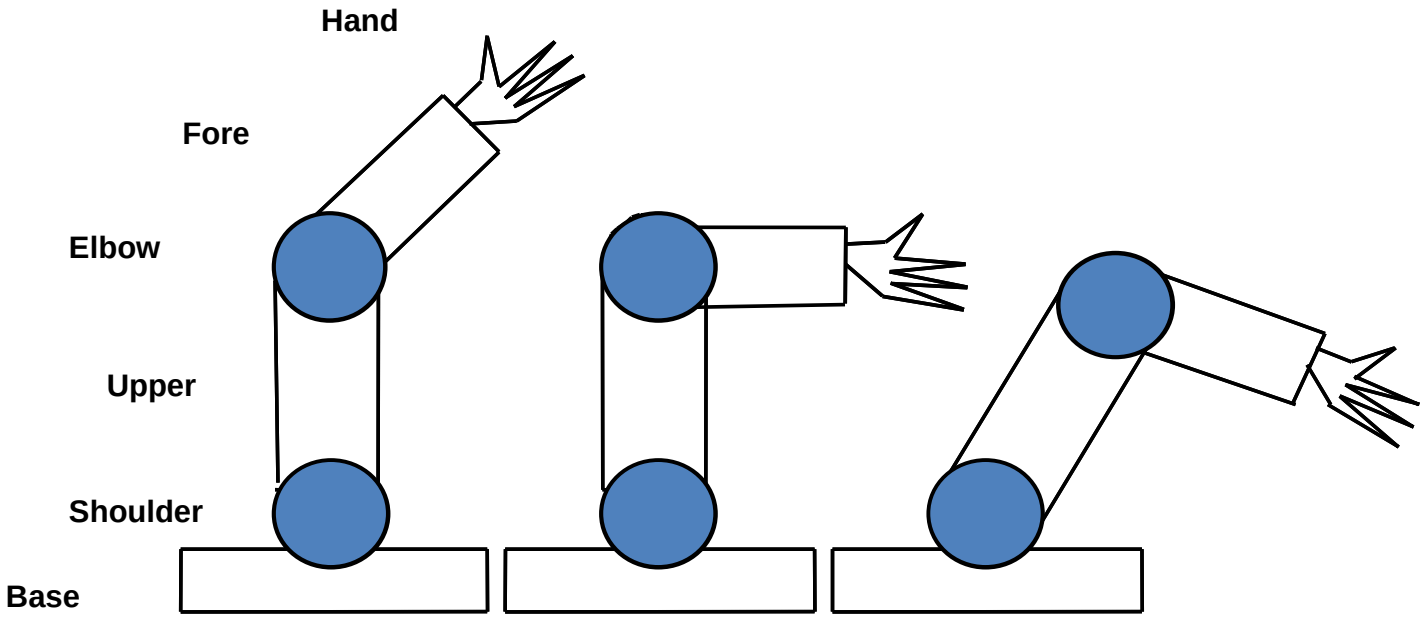
- Tree-like Scene Structure
  - Traversal 遍历
  - Instancing and Re-Use
  - More Transformations

Concept of Scene Graph

- **Objects placed relative to one another (LC)**
- Objects might be made of similar components
- Directed acyclic graph
- Internal nodes hold grouping and other information
- Links are transformations
- Leaf nodes contain geometry
- The root of the graph corresponds to “world coordinates”

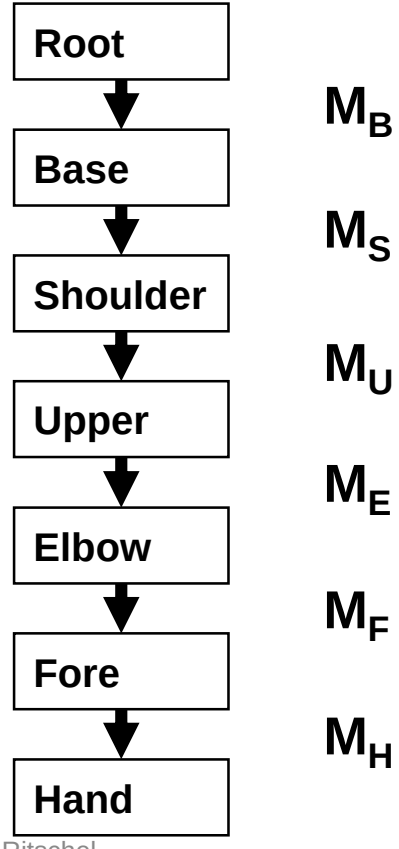


Use for Animation/Modelling



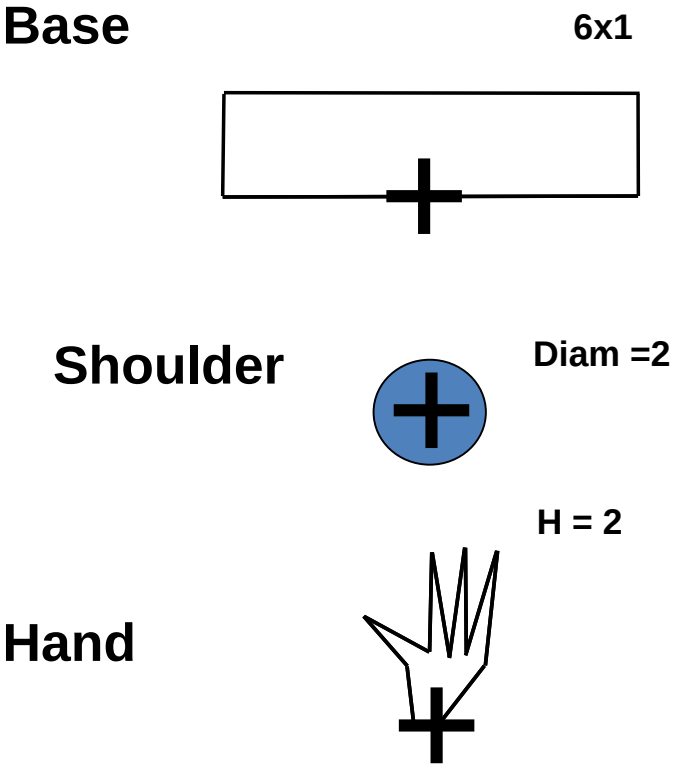
Robot as a Graph

- Each node other than root contains a piece of geometry
- Each link is a transformation matrix, $\mathbf{M}_B$, $\mathbf{M}_S$, etc.
- Main concept is that robot can be posed by changing rotation in Shoulder and Elbow



Local Coordinates

- Each part of the robot is modelled in its own local coordinate (LC) system
- Local coordinates are defined by the person modelling the system
- Choice is determined by convenience
- Common choices:
  - The centre of the object
  - A corner of the object



World Coordinates

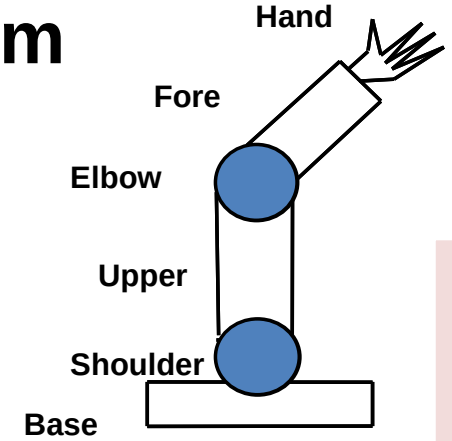
- Everything is eventually positioned relative to the world coordinates (WC) or room coordinates
- We know how to convert WC to viewing coordinates (VC) – it's the general camera model
- Eventually we need to convert points in an object's LC into WC

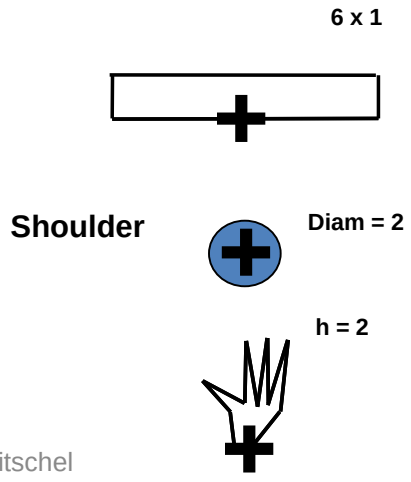
Local Transform

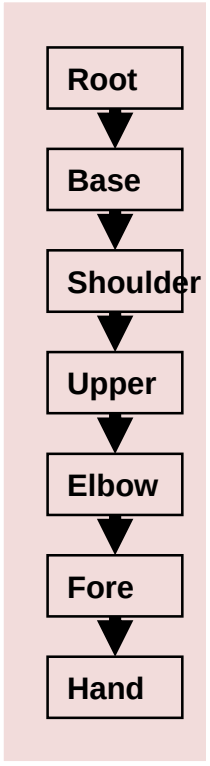
- An object's local transformation maps LC to the parent's LC
  - shoulder is translation (0 1 0) from base (M_S)
  - upper arm is translation (0 3 0) from shoulder (M_U)
  - elbow is translation (0 3 0) from upper arm (M_E)
  - fore arm is rotation Z by 90 then translation (0 2 0) (M_F)
  - Etc.
- Note that directions such as "up" depend on what transformations have been defined by ancestors in the tree

Local Transform

- An object's local transformation maps LC to the parent's LC
  - shoulder is translation (0 1 0) from base (M_S)
  - upper arm is translation (0 3 0) from shoulder (M_U)
  - elbow is translation (0 3 0) from upper arm (M_E)
  - fore arm is rotation Z by 45 then translation (0 2 0) (M_F)
  - Etc.
- Note that directions such as "up" depend on what transformations have been defined by ancestors in the tree





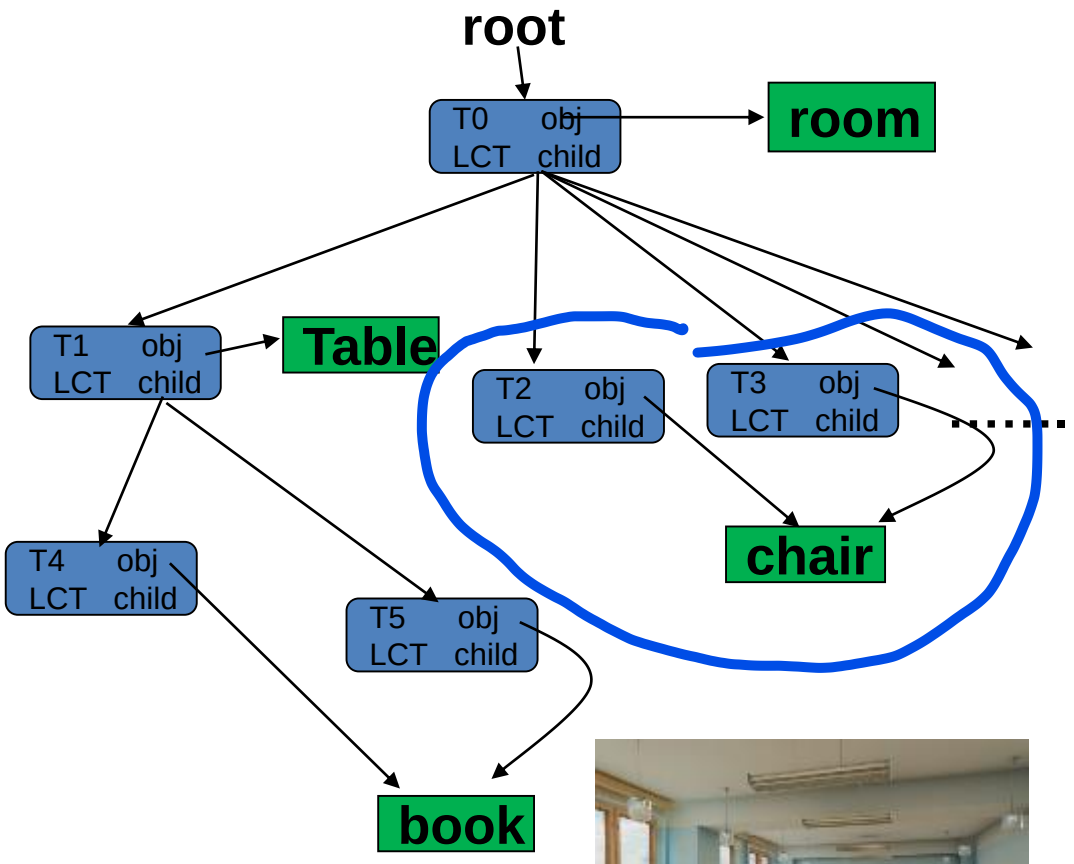


Example Scene

Sharing Nodes

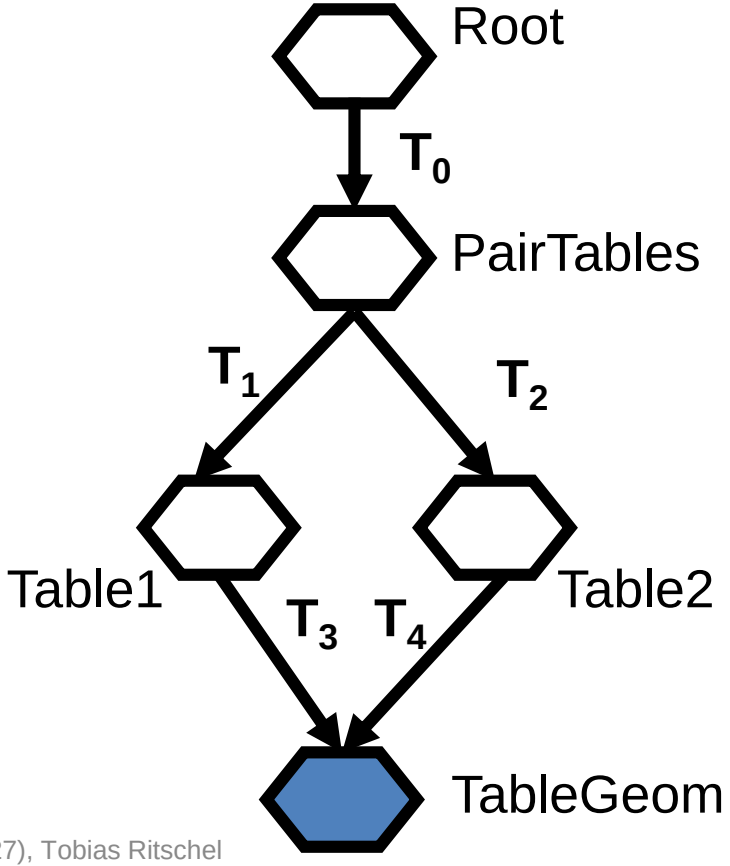
- E.g. A chair or table in many locations
- In practice

```
struct Node {
        Shape Object;
        Matrix CTM,
LTM;
        int
numChildren;
        Node child[];
}
```



Sharing Nodes

- A common “pattern” found in a scene graph is a multiple instanced geometry
- One table, many places
- Node Table1 has CTM $\mathbf{T}_1\mathbf{T}_0$
- Node Table2 has CTM $\mathbf{T}_2\mathbf{T}_0$
- $\mathbf{T}_3 = \mathbf{T}_4 = \mathbf{I}$
- So TableGeom appears in two different positions

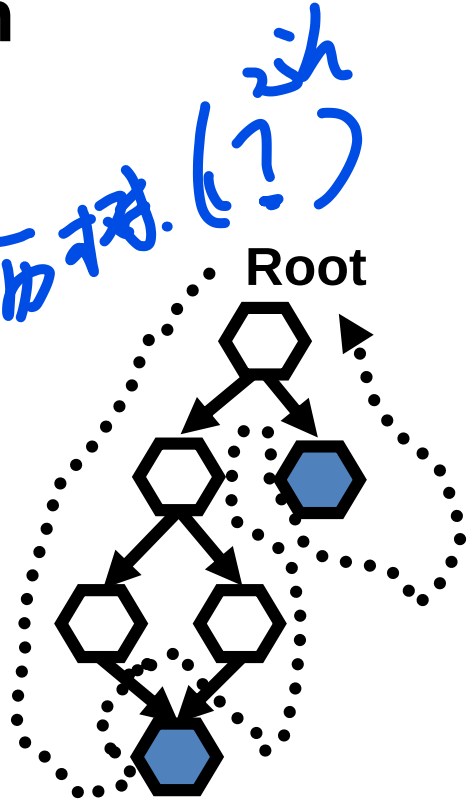


Rendering Traverse

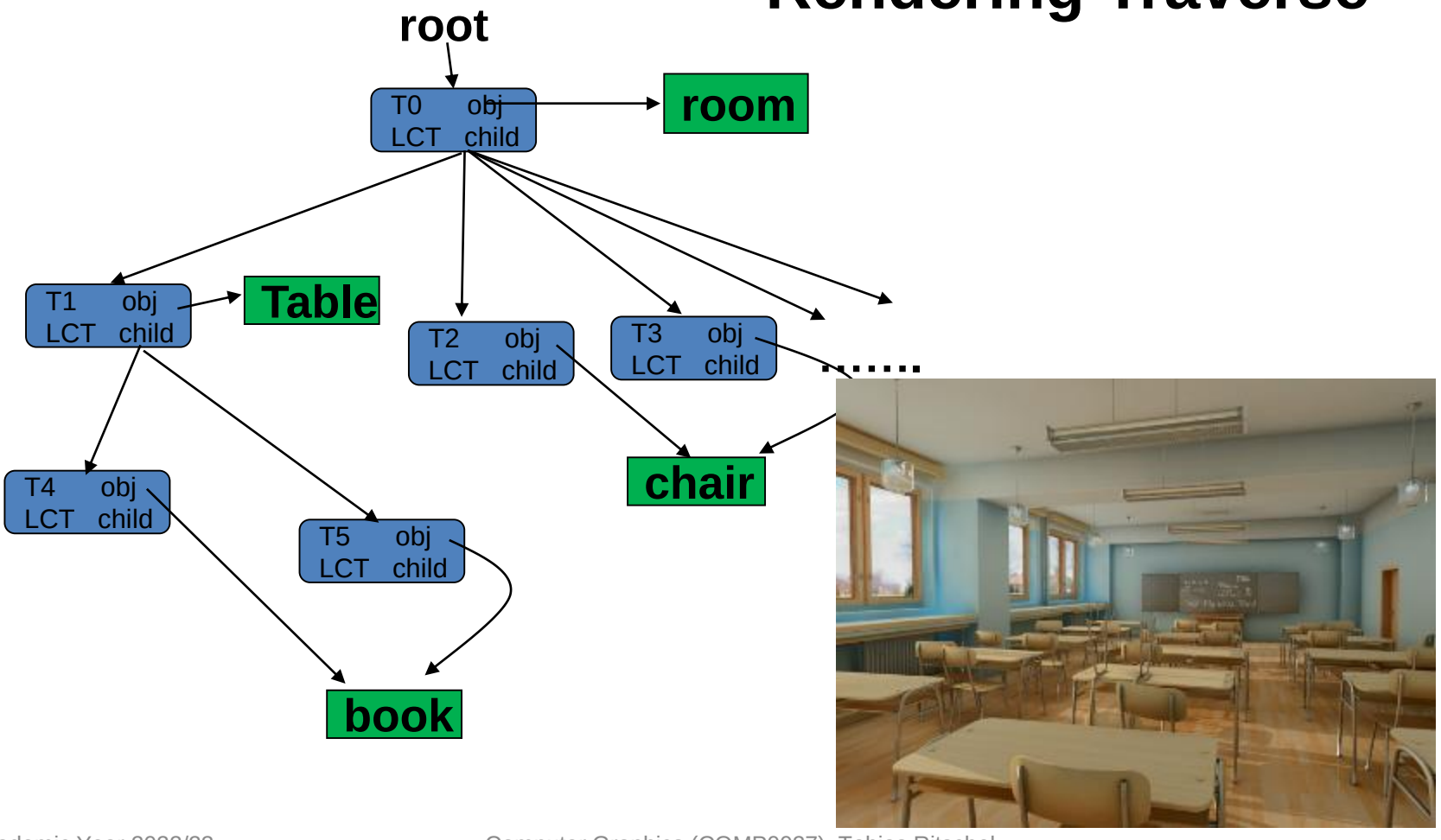
- Must get object definitions in WC before passing to camera
- For a vertex in the base object
  - p.$\mathbf{M}_B$ is in WC
- Matrices are inherited down stack
- So for object under shoulder
  - p.$\mathbf{M}_S\mathbf{M}_B$ is in WC
  - (Note that p.$\mathbf{M}_S$ is in the local coordinates of the base!)

Implementation

- Generally implemented by a straightforward recursive descent
  - "push" on graph descend
  - "pop" on graph ascend
- The concatenation of all LT matrices above a node is called the current transformation matrix (CTM)



Rendering Traverse



Recap

- Use of **scene graph** to model environments
- Local coordinates (LC) vs World coordinates
- Notion of render traversal and the **current transformation matrix**
- Instancing and sharing of nodes