Computer Graphics

Scene Graph

Andreas Aristidou
andarist@ucy.ac.cy
http://www.andreasaristidou.com

Scene Graph

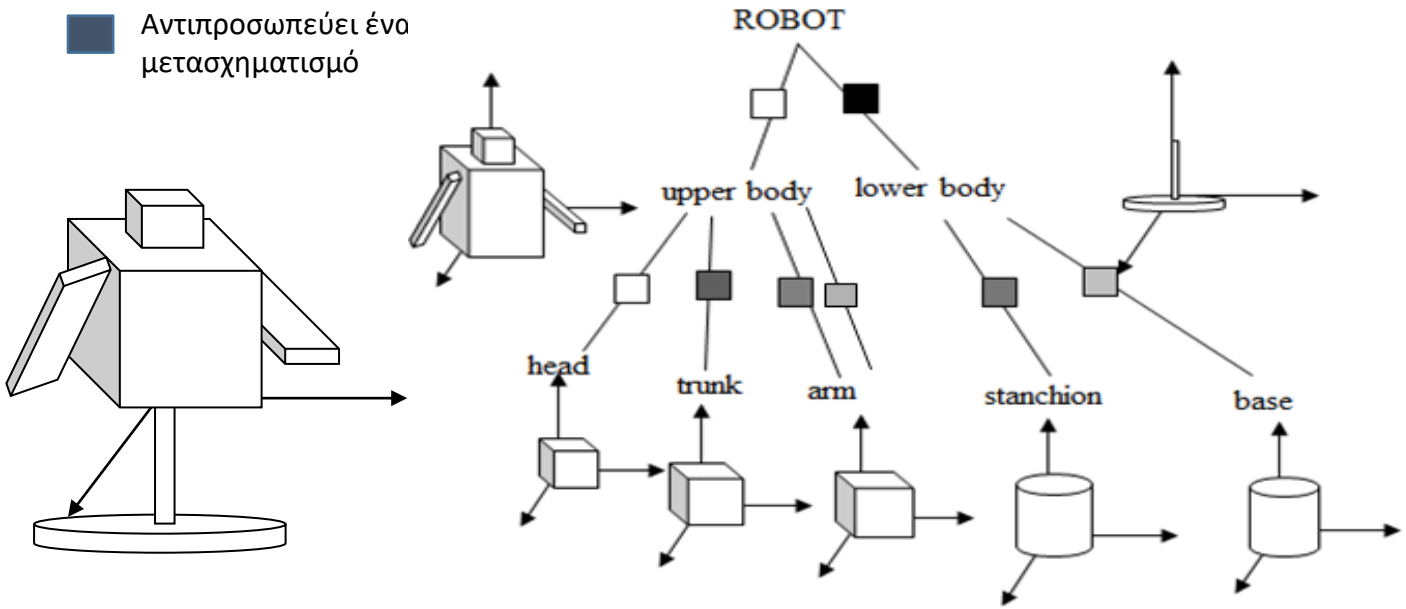
Scene Graph is a general data structure commonly used by vector graphics processing applications and modern computer games, which arranges the logic and often the spatial representation of a 2D/3D scene.
It is a collection of nodes in a graph or in a tree structure.

- A tree node can have multiple child nodes but only a single parent.
- A function that runs on a group of nodes, automatically transmits its effect to all its children.

The ability to group related shapes and objects in a **complex object**, helps to handle the group as easily as an individual object.

Transformations and the scene graph

- **Step 1:** Various transformations are applied to each of the leaves (e.g., head, base, etc.)
- **Step 2:** Then the transformations are applied to groups of objects (upper and lower body, etc...)

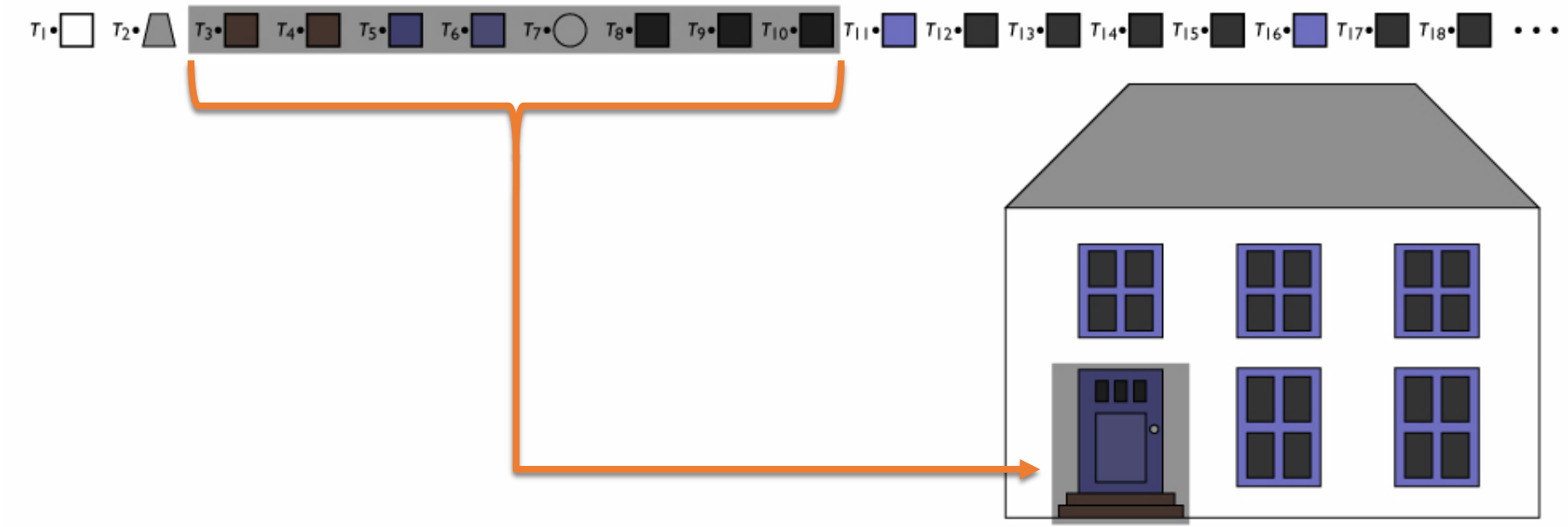


This format means that instead of designing new entities for each individual schema we need, we can simply apply transformations to a smaller set of entities to form complex shapes in 3D.

The above hierarchy of transformations together forms the "robot" as a whole

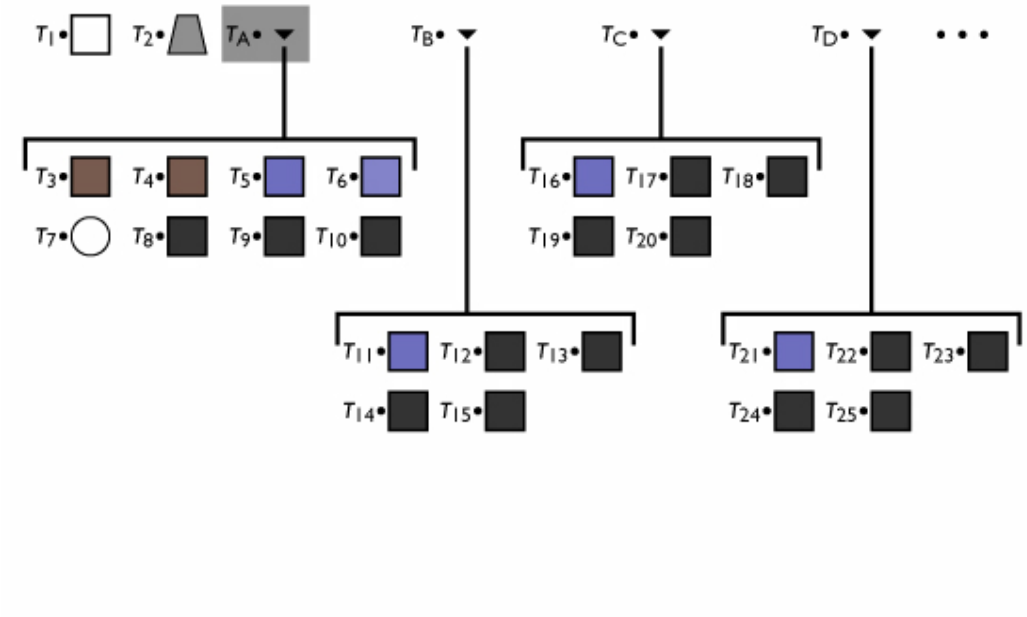
Example of a Graph

- Consider the following house
- It can be presented as a collection of objects
  - However, if we decide to manipulate it, it requires that we would use multiple transformations.



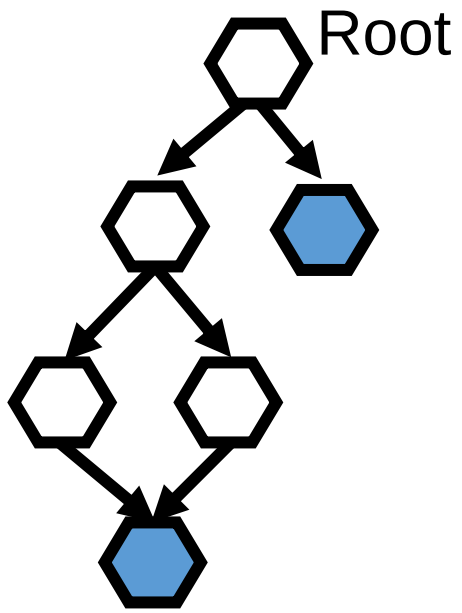
Grouping objects

- Create a group of objects as a new type of object
  - Allows the data structure to visualize the design structure
  - Allows high-level editing by changing only one node



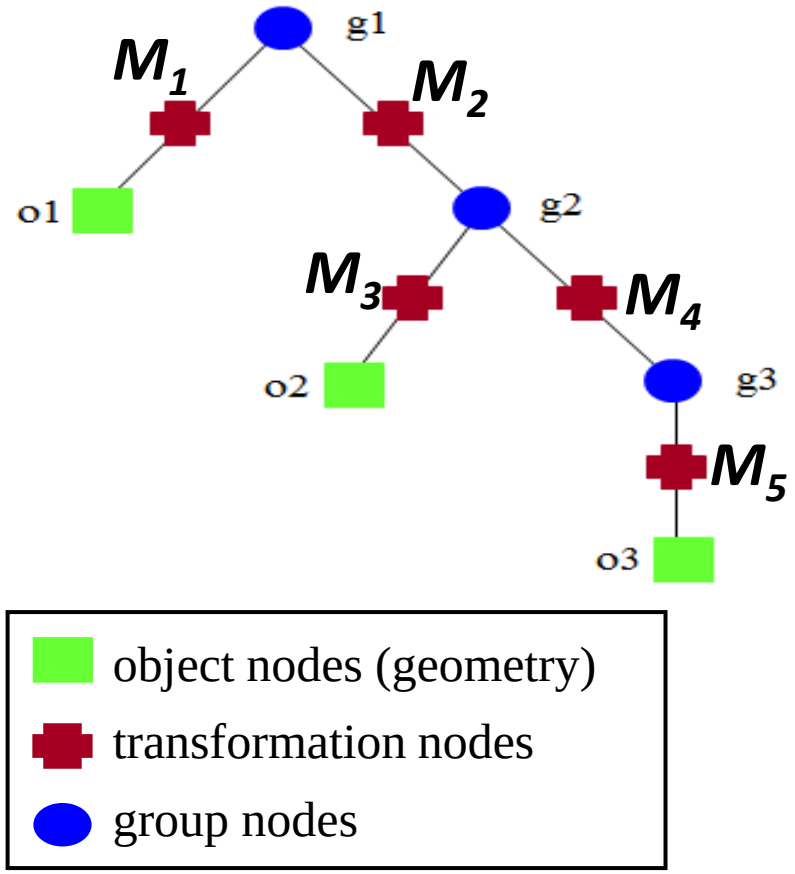
Meaning of the scene graph

- **Objects are placed in relation to each other (local coordinate)**
- Objects may be made of similar elements.
- Directed "acyclic" graph.
- Internal node grouping and other information.
- Connections are transformations.
- Leaf nodes contain geometry.
- The root of the graph corresponds to the "world coordinates“.



Transformations and the scene graph

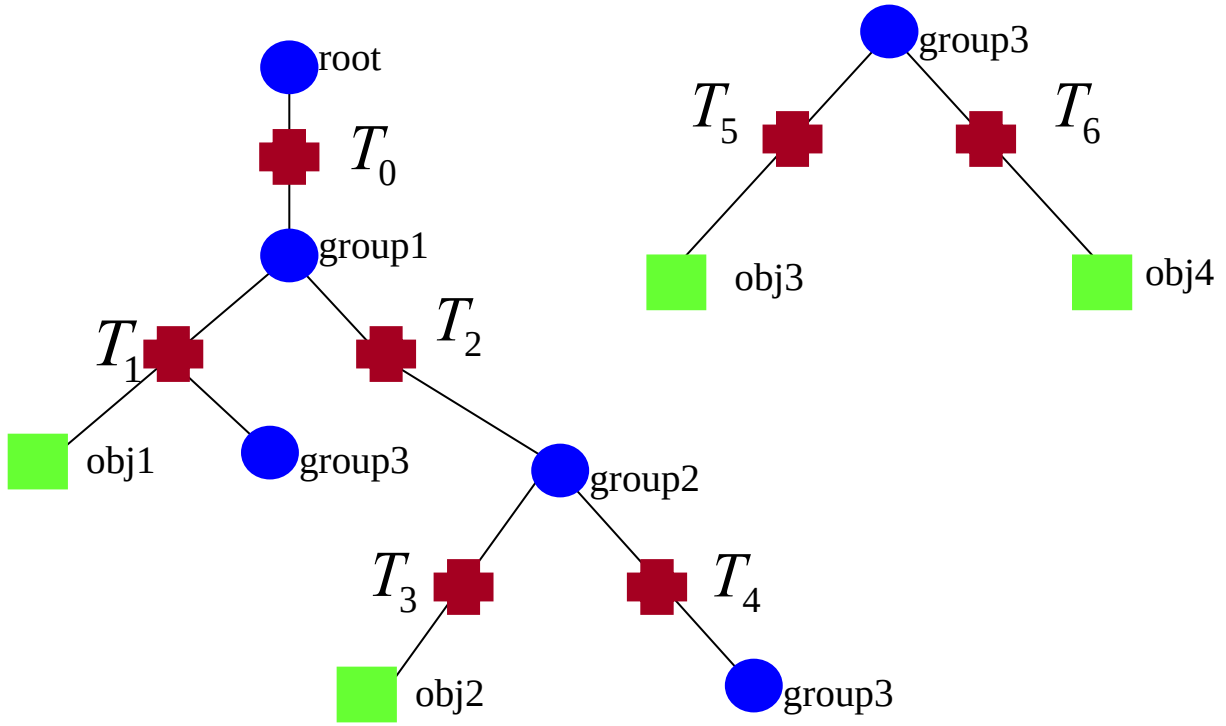
- A cumulative transformation matrix (CTM) is created as you move the tree.
- note that the higher-level transformation matrices are applied to the front of the sequence.



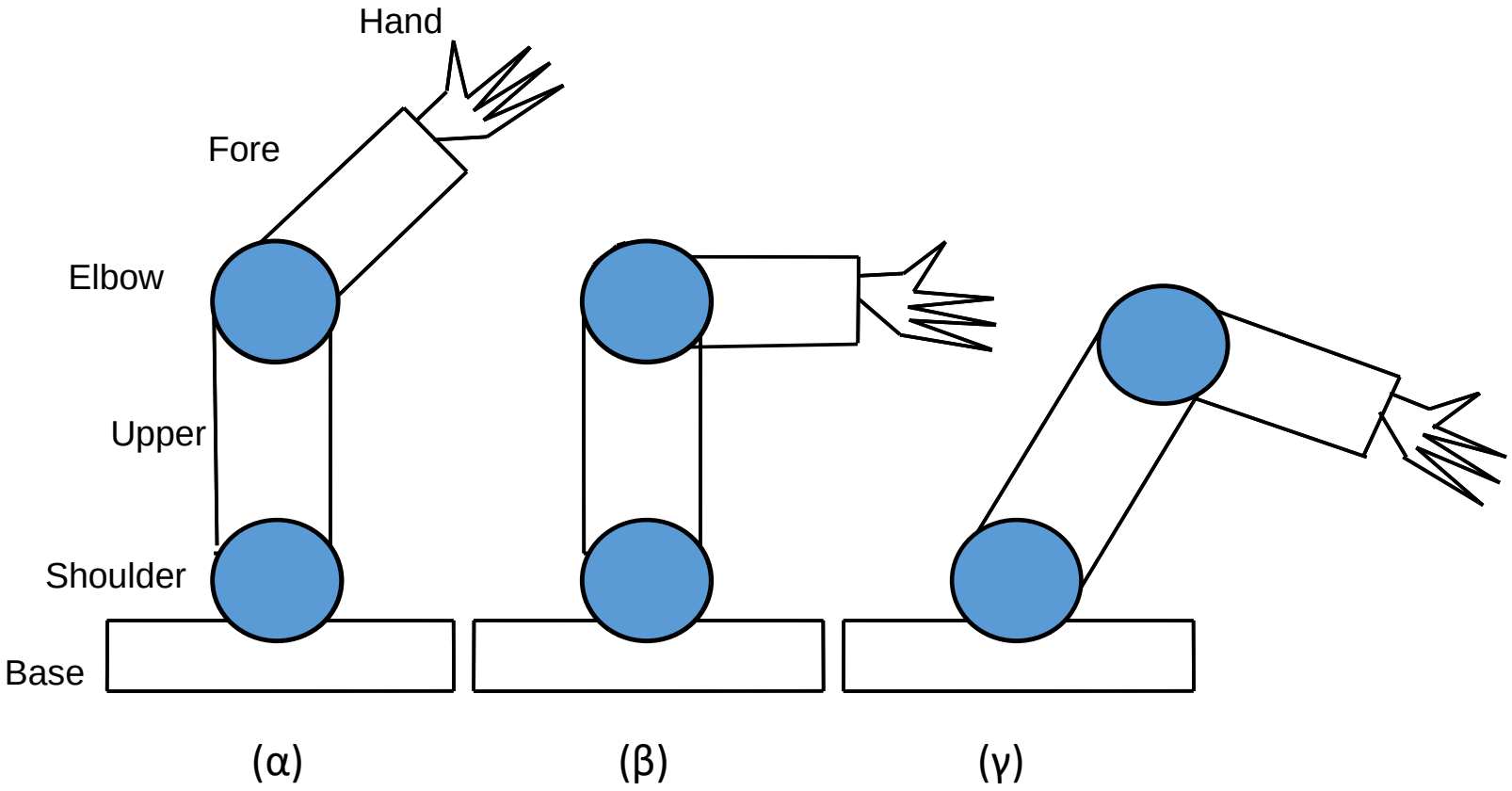
- Example:
  - For the object (o1), ***CTM*** = M_1
  - For the o2, ***CTM*** = M_2M_3
  - For the o3, ***CTM*** = $M_2M_4M_5$
  - For a vertex v in o3, the position in the coordinate system ***CTM*** is $v = (M_2M_4M_5)\,v$

Transformations and the scene graph

- You can easily reuse groups of objects (sub-trees in the scene graph) if they are already defined.
- This can happen if you have a lot of similar nodes in your scene. For example, the 2 hands of the robot.
- Here group 3 has been used twice.
- Transformations that are defined in group 3 itself do not change. There are different ***CTM***s for each use of group 3 as a whole.
- T_0T_1 vs. $T_0T_2T_4$

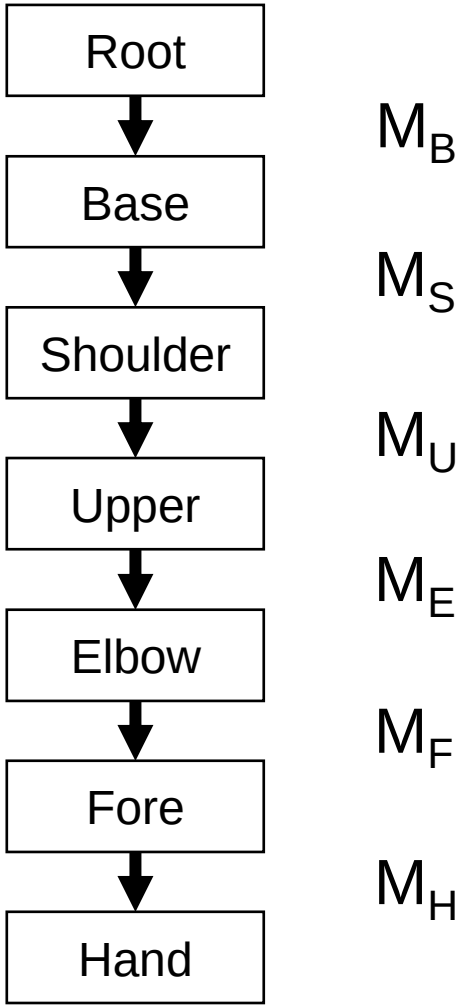


Use for animations/modeling



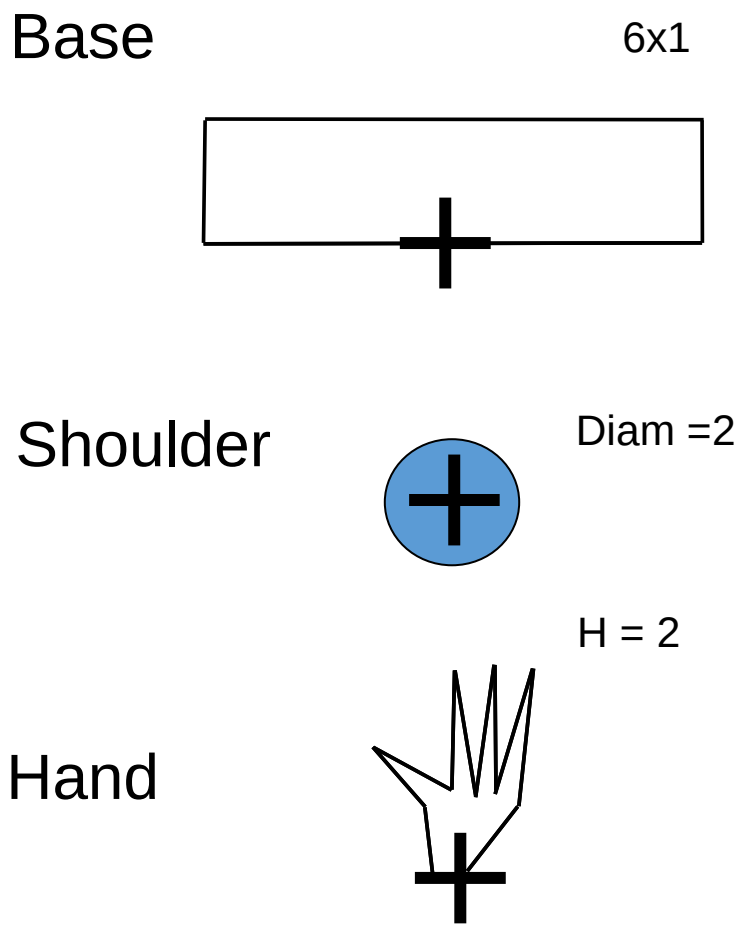
The robot as a graph

- Each node except the root contains a section of geometr.
- Each link is a transformation matrix, $\mathbf{M}_B$, $\mathbf{M}_S$, etc.
- The main idea here, is that the robot can move by changing the rotation on the shoulder and elbow.



Local Coordinates

- Each part of the robot is structured in its own local coordinate system (LC).
- Local coordinates are determined by the person who forms the system.
- The choice can be determined based on the **convenience of the creator**.
- Common options:
  - The center of the object
  - A corner of the object

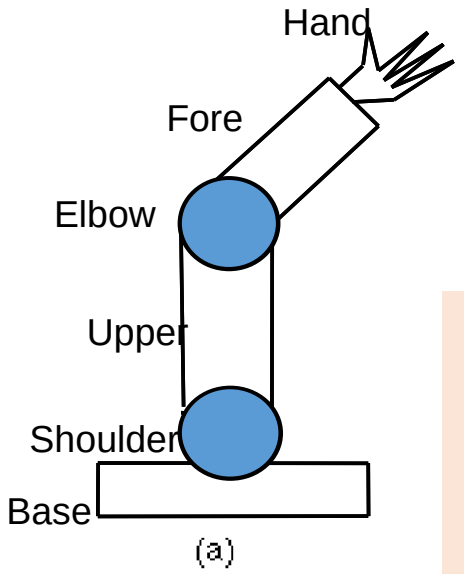


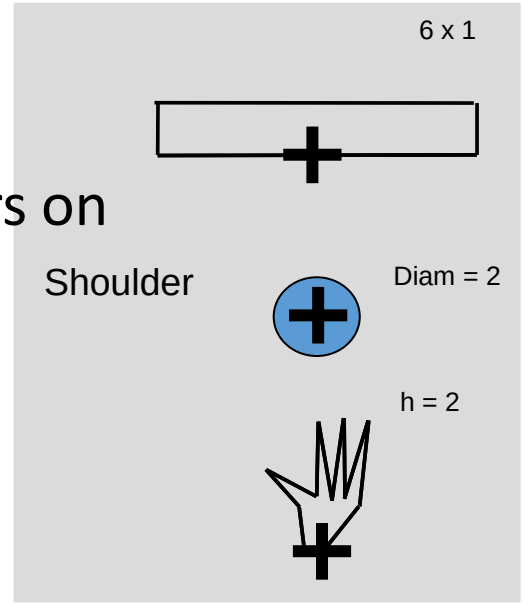
World Coordinates

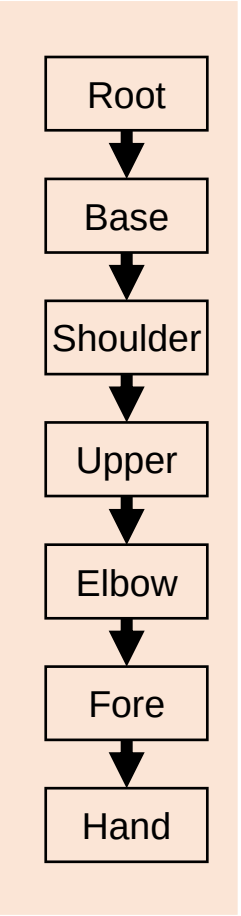
- All objects are eventually placed on the scene, relatively to the coordinates of the world (WC) or the room coordinates.
- Therefore, we must convert every point of the object from LC to WC.

Local transformation

- For the local transformation of an object, its LC corresponds to the parent's LC.
- The shoulder is translated by (0 1 0) from the base (M_S)
  - The upper arm translates (0 3 0) from the shoulder (M_U)
  - The elbow translates (0 3 0) from its upper arm(M_E)
  - The hand rotates on the Z axis by 45 and then moves (0 2 0) (M_F)
  - Etc...
- Note that directions such as "up" depend on what transformations have been defined by the ancestors on the tree.





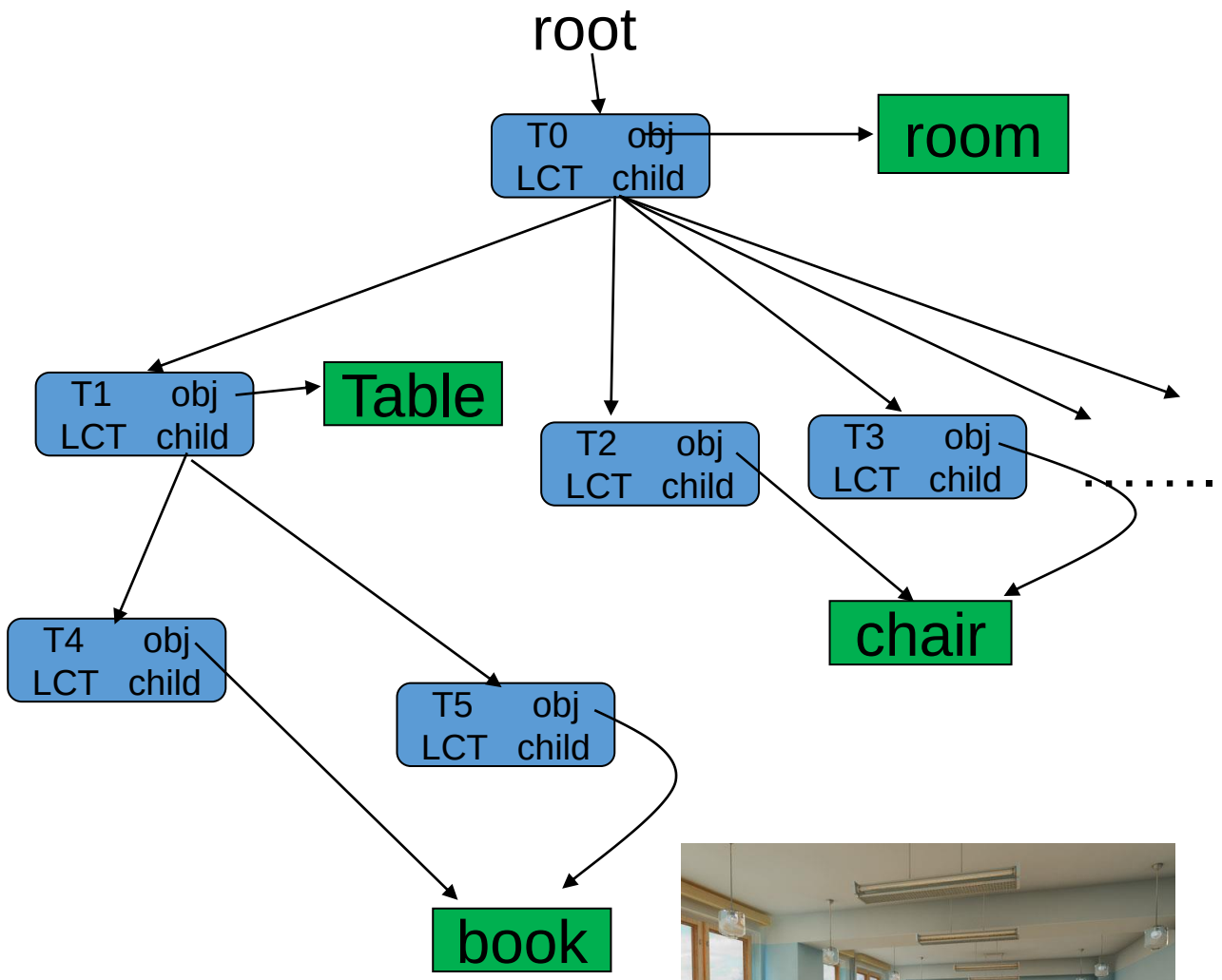



W/ MENTAL RAY

Share nodes

- E.g. chair or table in many locations

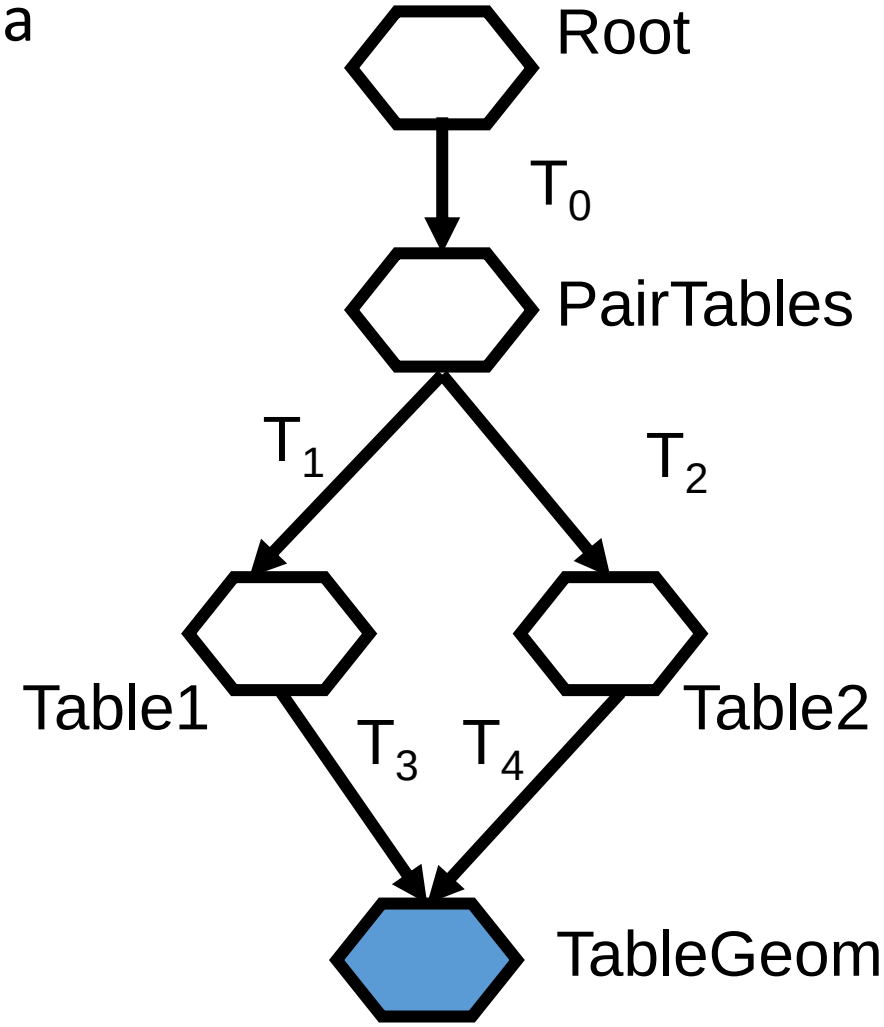
- In practice
- ```
  struct Node {
        Shape Object;
        Matrix CTM, LTM;
        int numChildren;
        Node child[];
  }
  ```




  ```

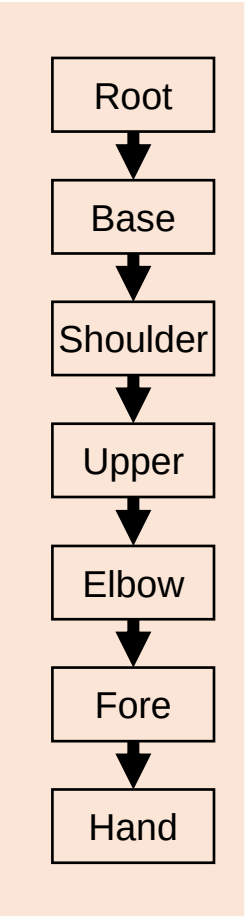
Share nodes

- A common "pattern" located in a scene graph is a geometry of multiple instance.
- One table, in many places
- The table1 node has a transformation CTM $\mathbf{T_0T_1}$
- The table2 node has a transformation CTM $\mathbf{T_0T_2}$
- $\mathbf{T_3 = T_4 = I}$
- So TableGeom appears in two different location.

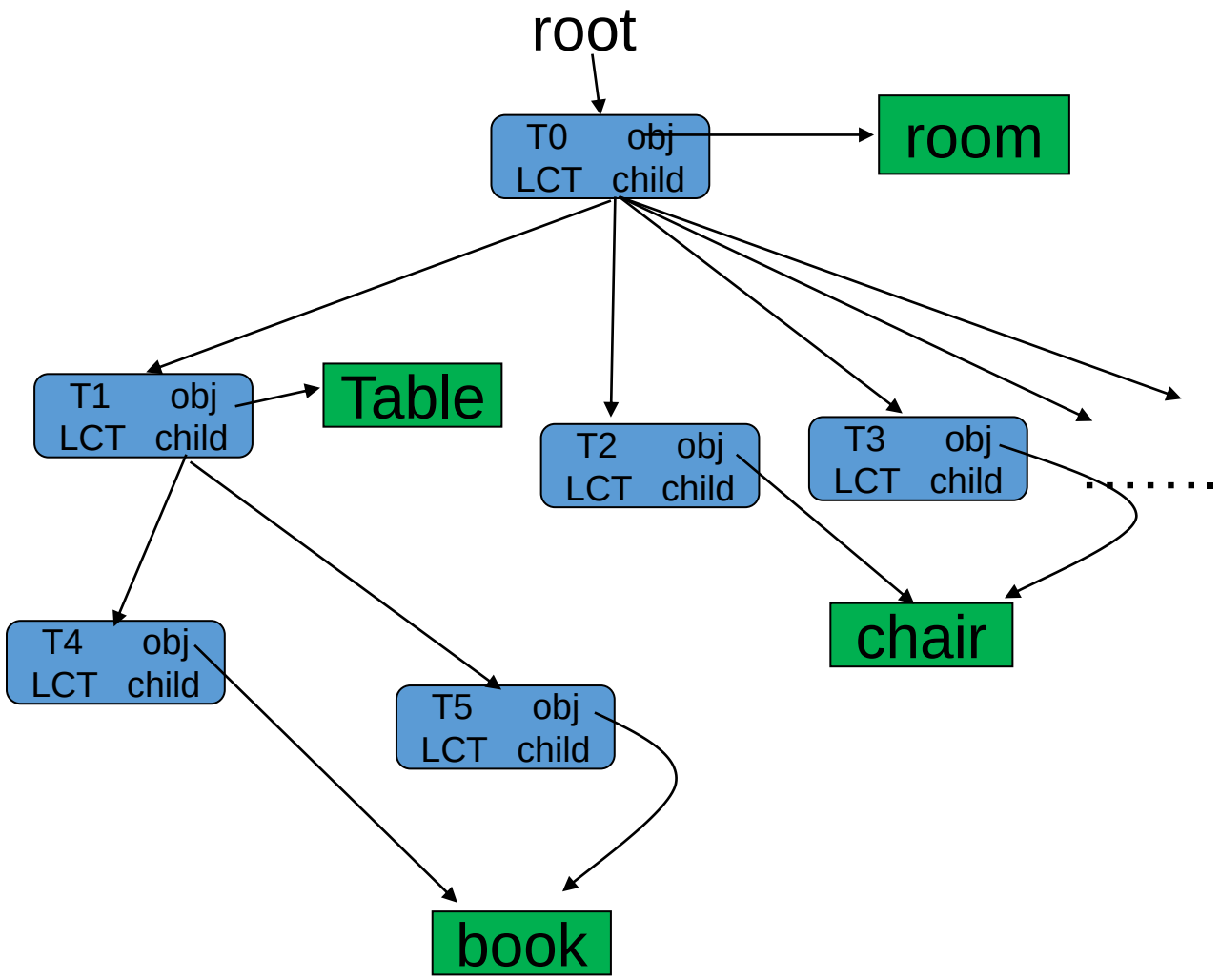


Rendering Traverse

- You need to take the positions of the objects in the WC before you start rendering, so they will be given to the camera
- For rendering we will talk later in the lesson
  - About converting from the coordinates of the world to the coordinates of the camera, we will talk in subsequent lessons



Rendering Traverse



Next lesson

- Graphic Modeling, Polygons, Polyhedra