Efficient Rendering

Learning objectives

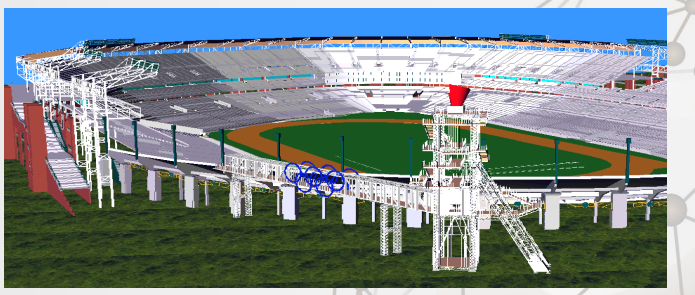
By the end of the module, you should be able to:

- Understand hierarchical representations
- Understand two forms of scene organization
- Construct and use bounding volumes
- Design and use level of detail (LOD)

1. Introduction

Problem: Geometric datasets can be too complex to render at interactive rates.

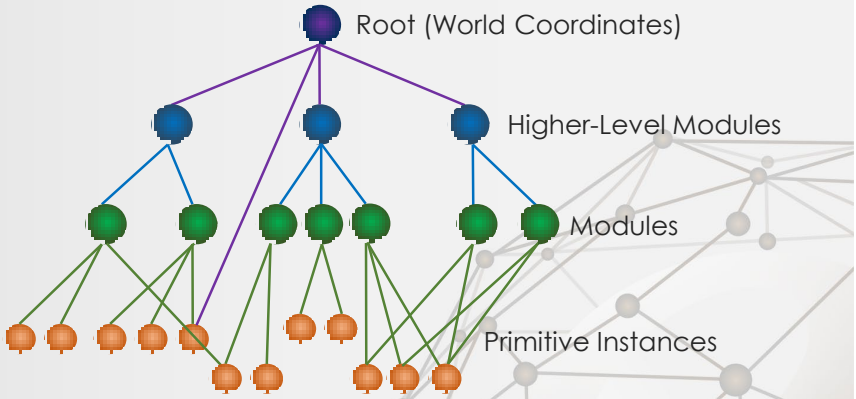
- The complexity is due to:
  - Too many objects
  - Too complicated representation of models



Techniques for fast rendering

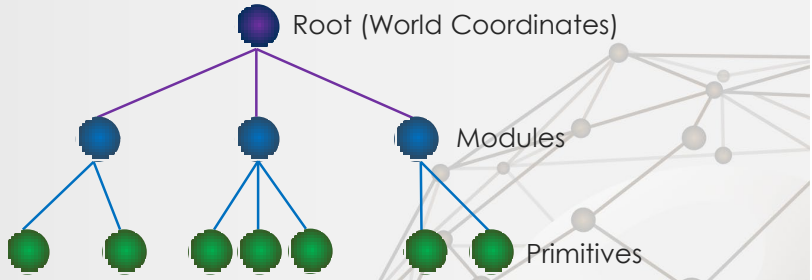
- There is possibility to speed up rendering by well-organizing data and smartly-processing data, in addition to just making use of advanced hardware. For example:
  - Hierarchical representation of models
  - Spatially-based organization of scenes
  - Bounding volumes
  - Level of detail

Hierarchical representation



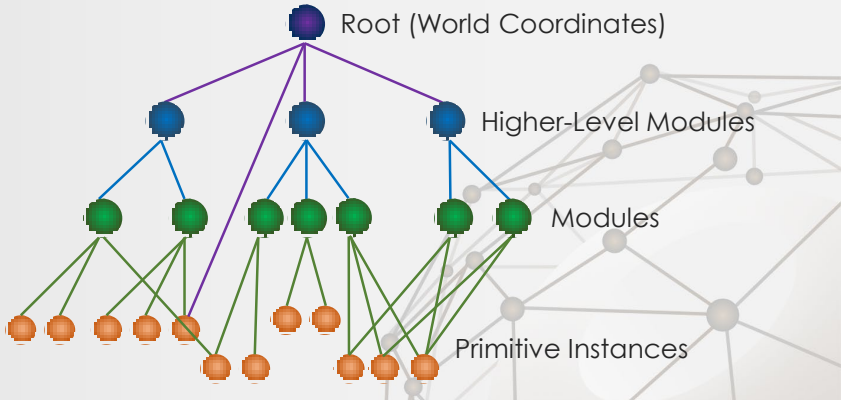
Hierarchy: Tree structure

- Tree: Each object is included only once in a higher-level object. This is not a common case.

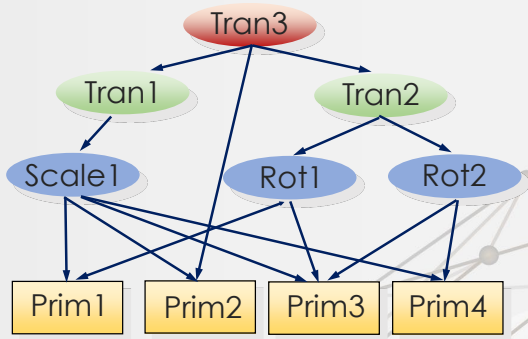


Hierarchy: Graph structure

- Directed acyclic graph (DAG): Objects are included multiple times. This is the common case.



An example of hierarchy with a DAG



It is also written algebraically as:

Tran3 (Tran1 (Scale1 (Prim1 + Prim2 + Prim3 +Prim4)) + Prim2 + Tran2 (Rot1 (Prim1 + Prim3) + Rot2 (Prim3 + Prim4))).

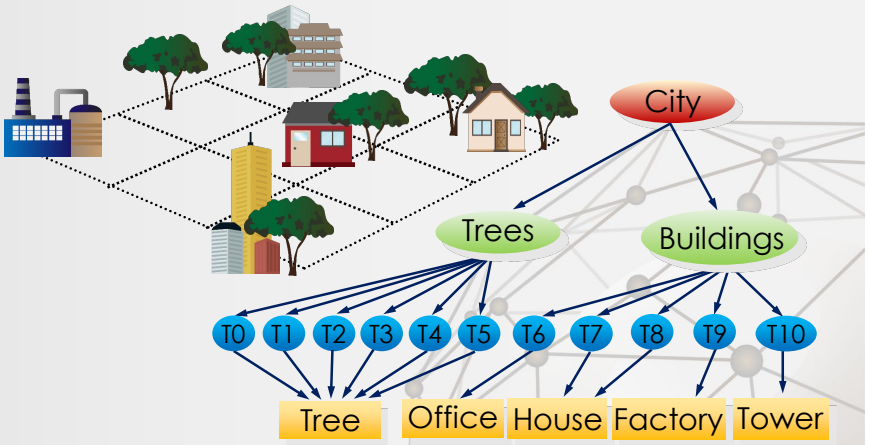
Advantage of hierarchy

- To construct complex objects in a modular fashion by repetitive invocation of objects that vary in geometric and appearance attributes.
- To increase storage economy, since it suffices to store only references to objects that are used repeatedly.
- To allow easy update propagation, because a change in the definition of one building-block is automatically propagated to all higher-level objects that use that object (like procedure hierarchy).

Therefore, hierarchy structure facilitates modeling, representation, and fast visualization as well.

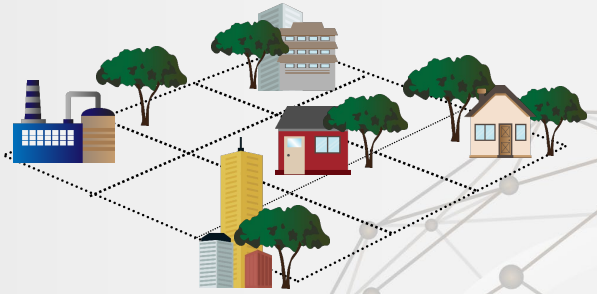
An example of logically-partitioned organization

- The organization is based on semantics of models.



3. Scene organization

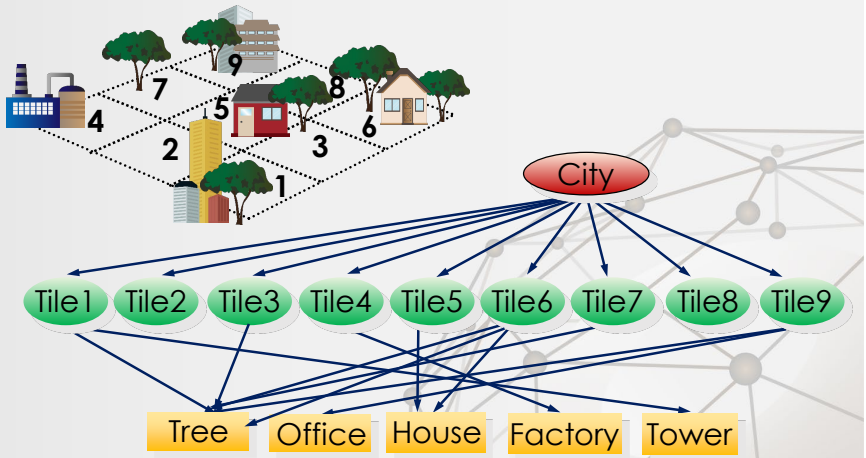
Problem: Given a scene consisting of many 3D models, as shown below, how to organize it (hierarchically)?



- Two approaches:
  - Logically-partitioned organization
  - Spatially-partitioned organization

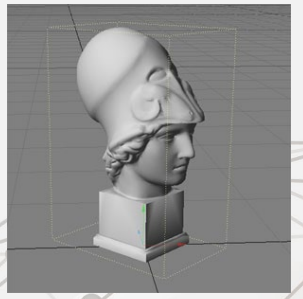
An example of spatially-partitioned organization

- The organization is based on the location of models.



4. Bounding volumes

- A bounding volume for a set of objects is a closed volume that completely contains the union of the objects in the set.
- Bounding volumes are used to improve the efficiency of geometrical operations by using simple volumes to replace complex objects.



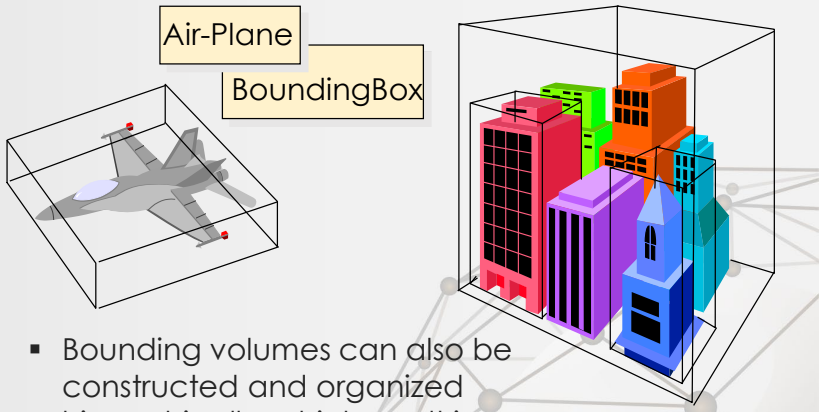
Factors for choosing a bounding volume

- There are a few factors that need to be considered in constructing bounding volumes:
  - The bounding volume should be relatively simple, geometrically. This makes the computational cost of geometrical operations with the bounding volume be low.
  - The bounding volume should bound the object tightly. This makes the use of the bounding volume for the object effective.
  - The computational cost of constructing a bounding volume for an object should be low.

Common types of bounding volume

- Bounding box
  - Axis-Aligned Bounding Box (AABB)
- Bounding sphere
- Bounding ellipsoid
- Bounding cylinder
- Discrete oriented polytope (DOP): A convex polytope resulting from the intersection of half-spaces bounded by the planes

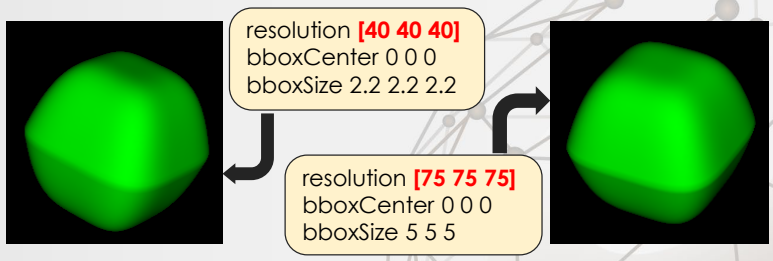
Examples of bounding box



- Bounding volumes can also be constructed and organized hierarchically, which result in bounding volume hierarchy.

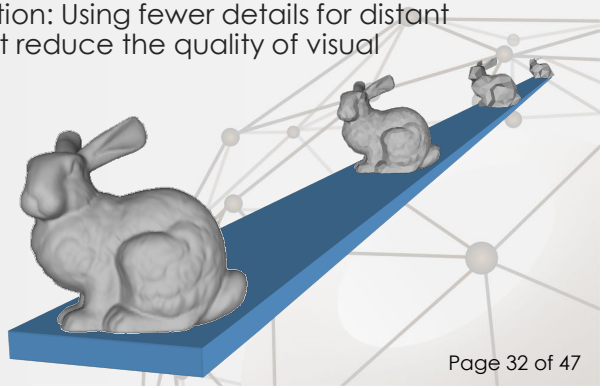
Bounding box in FVRML

- In FVRML, bounding box can be used to speed up the rendering of implicit objects and improve the rendering quality.
- For example, we use FVRML to visualize a shape defined by: $1-|x|-y^4-z^4 \geq 0$.
- To achieve similar rendering quality, we can use a lower resolution with a tight bounding box.



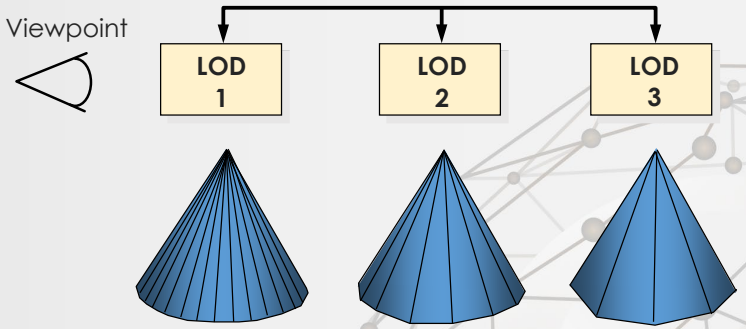
5. Level of detail

- Level of detail (LOD) is an important technique for **maintaining interactivity**.
  - Basic idea: Tradeoff among the fidelity of models, the visual fidelity, and performance.
  - Basic observation: Using fewer details for distant objects will not reduce the quality of visual appearance.



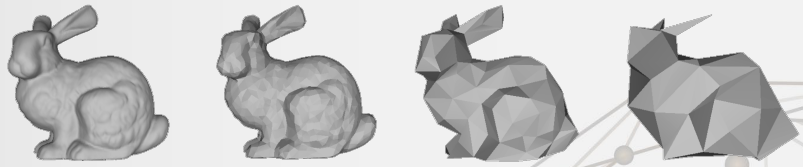
Basic rules of LOD

- Basic rules: More details for closer objects, and fewer details for farther objects.



Two basic issues

Issue 1: How to generate and represent simpler versions of a complex model?

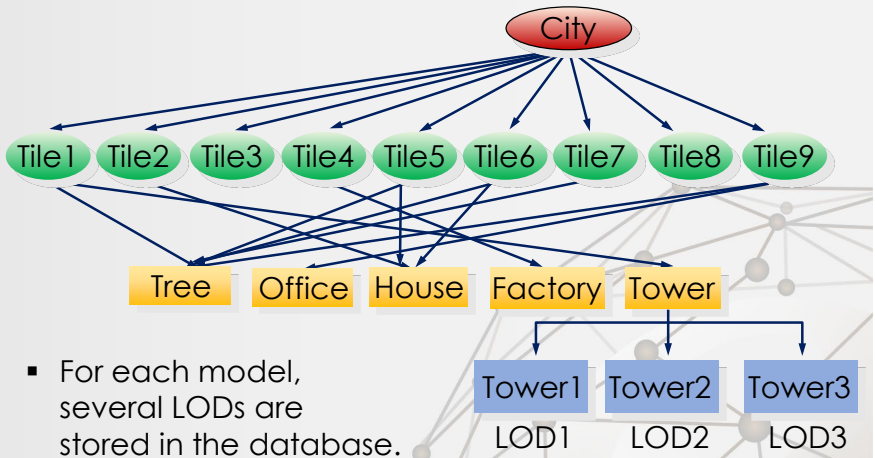


Issue 2: When to use which LOD of the model?

Method 1: Discrete LOD

- Create LODs for each object separately in a pre-process. Thus, LODs are created offline at fixed resolutions.
- At run-time, pick each object's LOD according to the object's distance or a similar criterion.

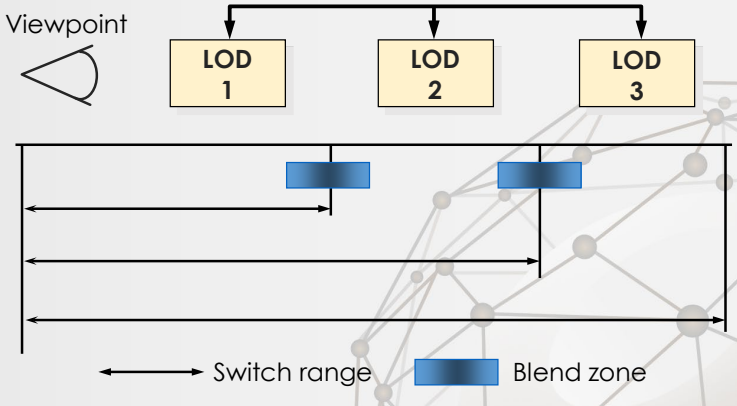
Store LODs in database



- For each model, several LODs are stored in the database.

Blending between LODs

- When the system switches between LODs, blending of LODs can be used to avoid discontinuity.



Method 2: Continuous LOD

Idea: Create data structure from which a desired level of detail can be extracted **at run-time**.

- Approach for implicitly- or parametrically-defined objects:
  - Make the resolution be function of the object's distance.
- Advantages:
  - LOD is specified exactly, not chosen from a few pre-created options.
  - Objects use no more polygons than necessary, which frees up polygons for other objects.
  - Smoother transitions, reducing visual pops.

Example: Creating continuous LODs

Q: Propose a method that implements continuous LODs for an origin-centered unit sphere.

Ans:

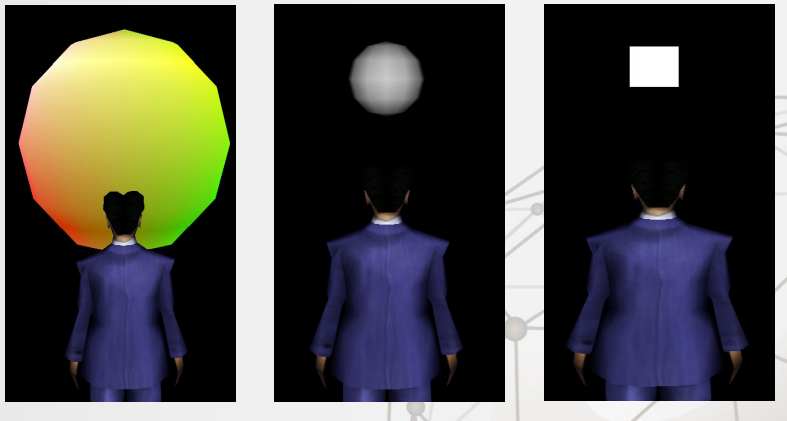
- Express the sphere in parametric form:

$$x = \cos\phi \cos\theta,$$
$$y = \cos\phi \sin\theta, \quad \pi/2 \le \phi \le \pi/2$$
$$z = \sin\phi, \qquad 0 \le \theta \le 2\pi$$

- Sample the surface with $\phi = -\frac{\pi}{2} + \delta \cdot i$, $\theta = \delta \cdot j$, where $\delta = \delta(d)$ is an increment constant and d is a distance from the observer to the sphere. The less d is, the smaller δ should be. For example, $\delta = c*d + \varepsilon$ where c and ε are two constants.

LOD in VRML

- In VRML, LOD node specifies various levels of detail or complexity in terms of colors and shape.



6. Summary

- To achieve fast rendering, one should reduce the number of models and computation amount.

- The following methods can be used:
  - Using modules and hierarchy representation
  - Spatial organization of scenes
  - Bounding volumes
  - Level of detail