## Other 3D spatial representations: Voxels

Voxels – divide 3D space into uniform cubes, like pixels divide 2D into squares

This way of representation has many drawbacks

- voxels will be very numerous especially for game levels set outdoors;
- need inside/outside distinction, containing object information
- for exterior voxels, colour and/or texture info needed for rendering;
- much data shuffling will occur when objects move;
- granular for both motion purposes and for visuals (cf. anti-aliasing in 2D)



## Other 3D spatial representations: octrees

Octrees – like quadtrees, each node either is a leaf or has eight daughters

coordinate ranges should ideally be power of two in all 3 directions
index daughter node by most significant bits of 3 coordinates, then leftshift them

- eg with 32x32x32 cube, a coordinate at <12,22,17> will be in
  <01100, 10110,10001> (ie division 011, decimal 3) then
  <01100, 10110, 10001> (its subdivision 100, decimal 4) then
  <01100, 10110, 10001> (its subdivision 110, decimal 6) … and so on

empty regions may be represented by vastly fewer nodes than voxels
(parts of) objects listed as occupying octree node must have colour/texture
issue: how does octree structure change as objects move?

- will existing cells be split as multiple objects are found to be within?
- will 7/8ths empty octets be simplified?



## Other 3D spatial representations: portals

Portals – doors, windows, turns along an outdoor pathway, level advancement

- 3D world is divided into "rooms" with portals connecting adjacent rooms
- A camera in a room sees inside that room; into adjacent ones through its portals
- Portals (e.g. doors) may be open or shut
  - see through open ones only
  - portal itself occupies both rooms
- Further portals may be seen through, and still further ones through them
- Much of what is in an adjacent room may be culled, not painted at all

Issues:

- what room is camera in?
  - can camera move without passing through portals?
  - can viewer go through walls, or teleport? then hunt the camera!
- potentially very many portals
  - consider an outdoor scene with skyscrapers: is each window a portal?



## Other 3D spatial representations: Potentially Visible Sets

Like with portals, space is divided into rooms (or "nodes")
Each room holds a list of others that might be seen from any point within it

- Conceptually simple
- For rendering, offers rapid elimination of many polygons that cannot be seen

Issues:

- Inherently conservative: a room **B** must always be in PVS of room **A**
  - if a door might be opened, regardless of whether it is open now
  - if **B** may be seen from *somewhere* in **A**, regardless of where camera is now
- Changing geometry (if doors or walls might get destroyed) is hard to handle
- Automatic generation of PVS is not easy
  - must every possible camera position be considered? if not …
  - geometry of room and/or camera position probably must be constrained