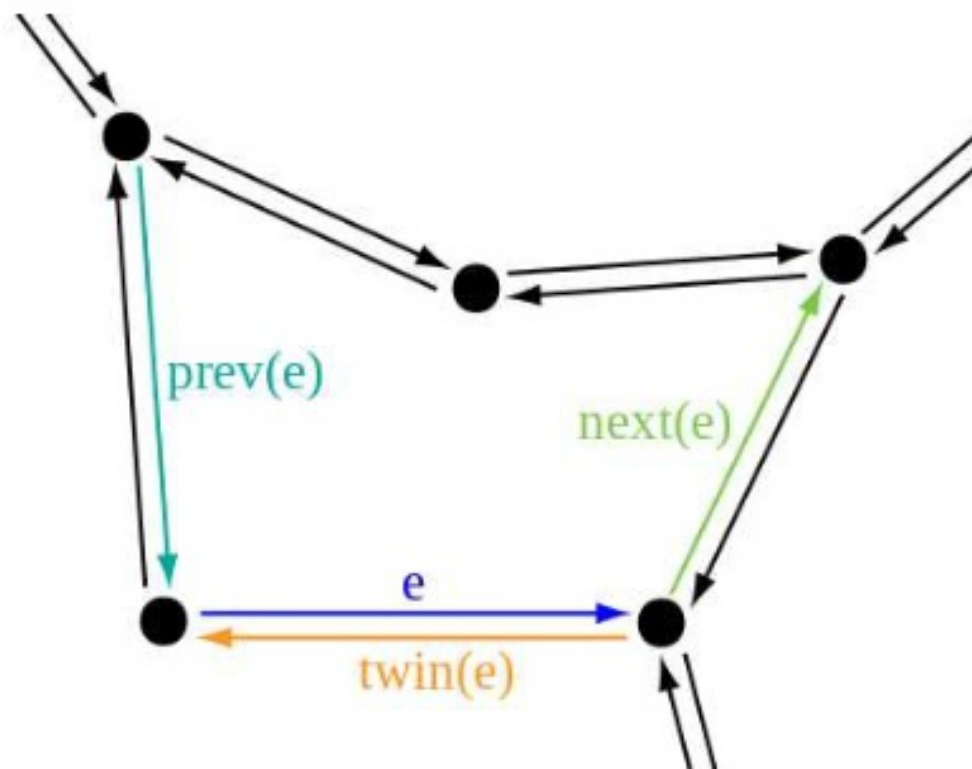


Chapter 3

The Geometry of Virtual Worlds

3.1 Geometric models

- 3D Euclidean space w/ Cartesian coordinates
 - let R^3 denote real world, using (x,y,z)
- **Data Structures**
 - Geometric models usually encoded in clever data structures
 - **Doubly connected edge list__ aka __Half-edge data structure**
 - Three kinds of data elements: *faces*, *edges*, and *vertices*
 - represent 2, 1, and 0-dimensional parts of model



-
- Figure 3.3: Part of double connected edge list shown for face w/ five edges on boundary. Each half-edge structure e stores pointers to the next and prev edges along face boundary. Also stores pointer to its twin half-edge, which is part of boundary of adjacent face)

- **Inside vs. outside**
 - Q : is object interior part of model?
 - **Coherent model**: If model inside were filled w/ gas, could not leak

- **Polygon soup:** Jumble of triangles that do not fit together nicely, could even have intersecting interiors
- **Why triangles?**
 - Triangles used because simplest for algorithms
- **Stationary vs. movable models**
 - Two kinds of models:
 - i. **Stationary models:** keep same coordinates forever
 - ex: streets, floors, buildings
 - ii. **Movable models:** can be *transformed* into various positions and orientations
 - ex: vehicles, avatars
 - Motion can be caused either by
 - i. tracking system (model match user's motions)
 - ii. controller
 - iii. laws of physics in virtual world
- **Choosing coordinate axes**
 - Don't be stupid.
- **Viewing the models**
 - *Q: How is model going to "look" when viewed on display?*
 - Two parts:
 - i. Determining where points in virtual world should display
 - ii. How each part of model should appear after lighting sources and surface properties defined in virtual world

3.2 Changing Position and Orientation