

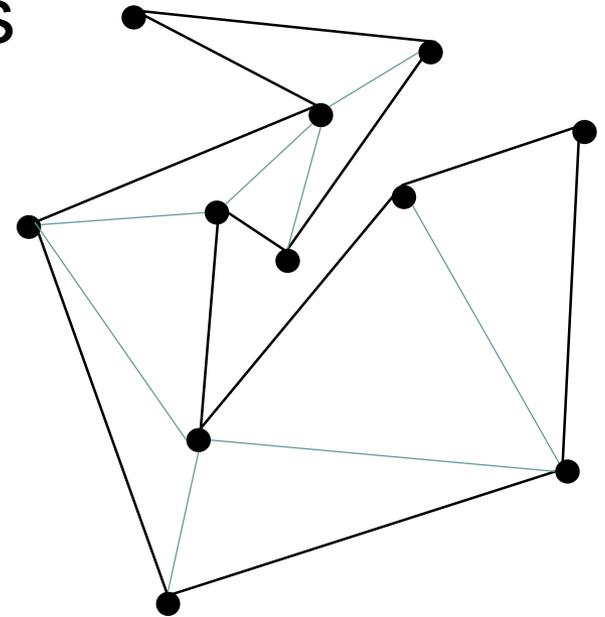
CS133

Computational Geometry

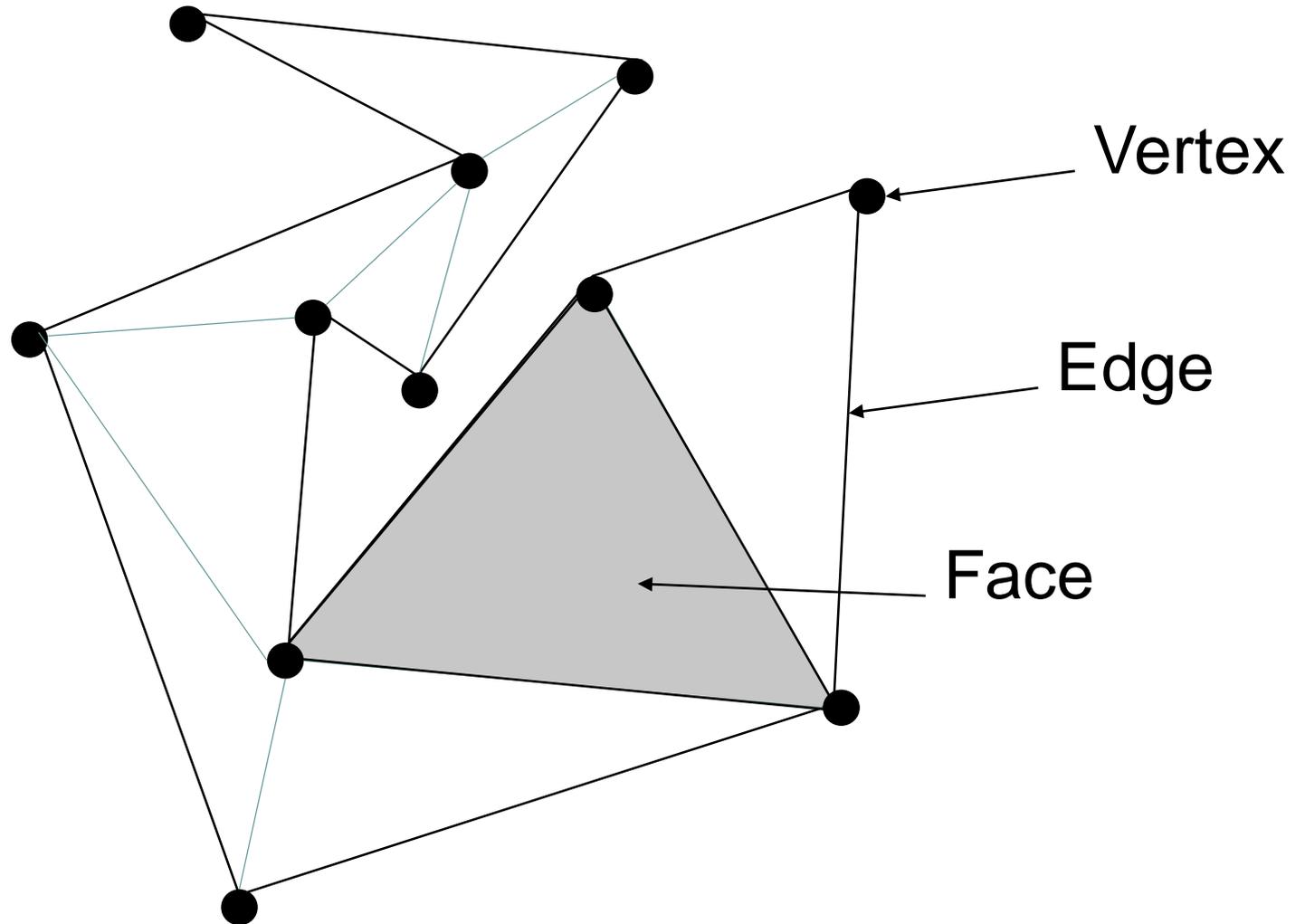
The Doubly-Connected Edge List
(DCEL) Data Structure

DCEL

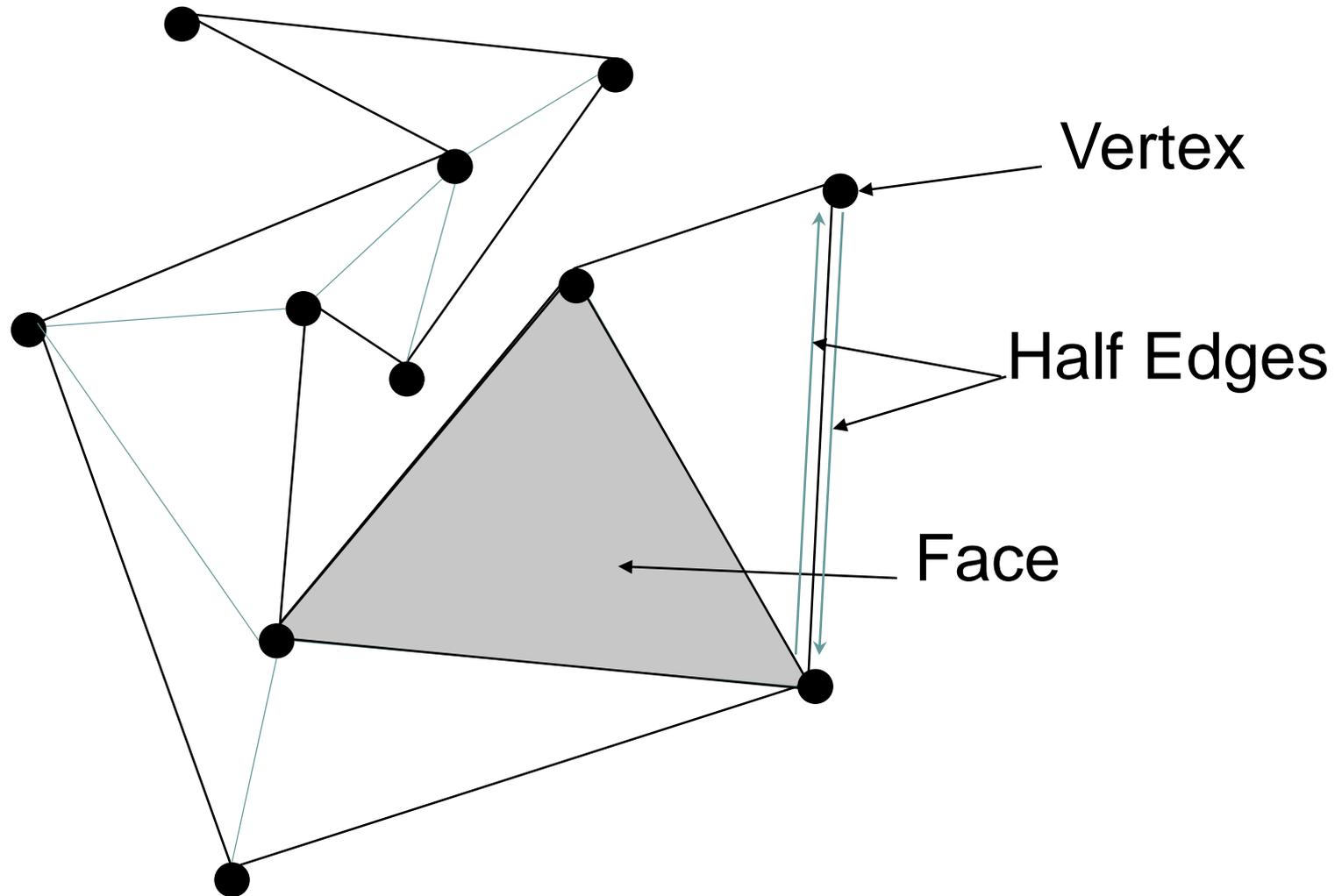
- DCEL stands for Doubly-Connected Edge List
- It is a data structure for representing topological information about a planar graph
- It also has some applications in representing 3D meshes



Terminology



Elements of DCEL



1- The Vertex Object

- Each vertex stores a single pointer to a HalfEdge that is leaving this vertex
- A vertex stores the coordinates of this point (not needed in other applications when only topology is required)
- Vertex {
 - HalfEdge* leaving;
 - Double x, y;
- }

2- The HalfEdge Object



- ▶ The HalfEdge contains pointers to
 - ▶ The origin Vertex
 - ▶ The Face that is incident to the left of the HalfEdge
 - ▶ The twin HalfEdge that points to the other half of this edge (the one to its right).
 - ▶ The next HalfEdge that originates from the destination of this HalfEdge and is incident to the same face

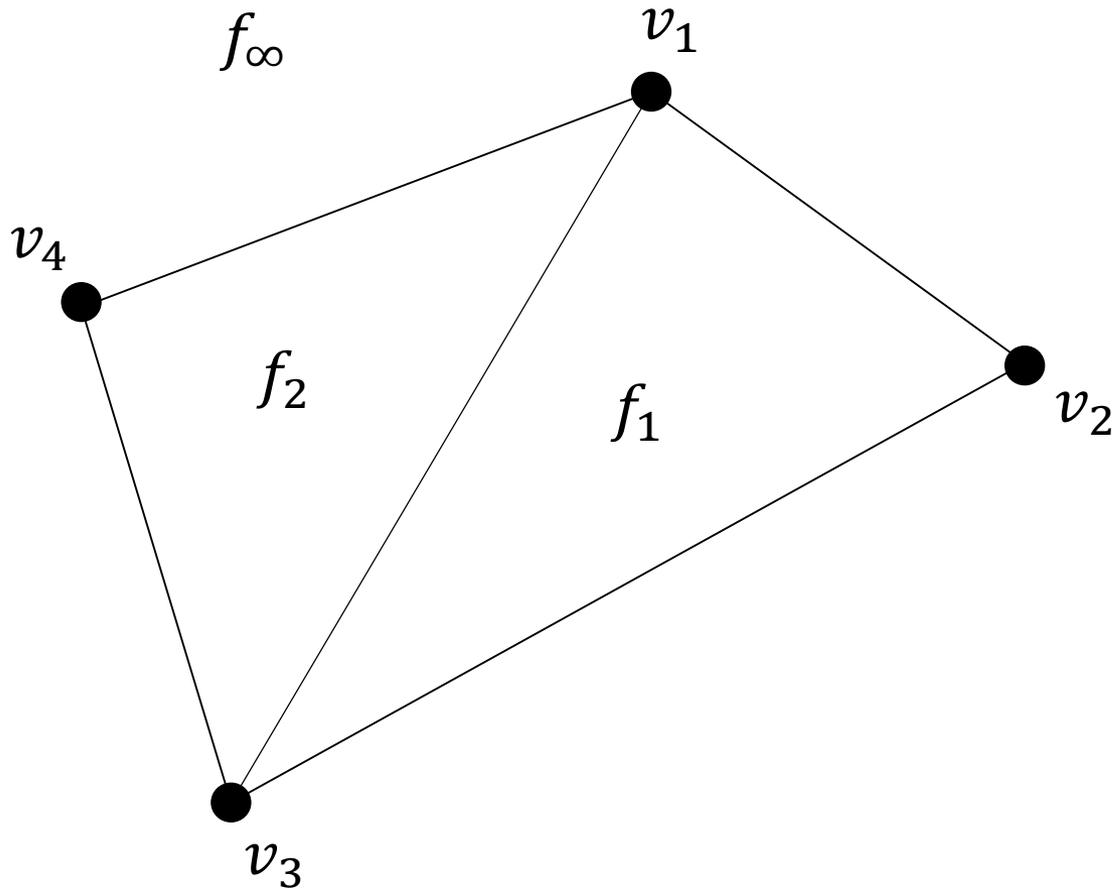
2- The Half Edge Object

- ▶ HalfEdge{
 - ▶ Vertex* origin;
 - ▶ Face* face;
 - ▶ HalfEdge* twin;
 - ▶ HalfEdge* next;
- ▶ }

3- The Face Object

- A Face contains a single pointer to an incident HalfEdge
- While a Face can be enclosed in many HalfEdges, only a single pointer to one of them is needed
- Face {
 - HalfEdge* edge;
- }
- To keep the structure consistent, we store a special *infinite* face

Example



Traversals

- ▶ Origin: Given a HalfEdge (e), find its destination
 - ▶ $e \rightarrow \textit{twin} \rightarrow \textit{origin}$
- ▶ NextLeaving: Given a vertex (v) and a HalfEdge (e) leaving v , find the next leaving HalfEdge in CW order
 - ▶ $e \rightarrow \textit{twin} \rightarrow \textit{next}$