600.657: Mesh Processing

Chapter 2

Data Structures

- Polygon/Triangle Soup
- Indexed Polygon/Triangle Set
- Winged-Edge
- Half-Edge
- Directed-Edge

Polygon Soup

- List of faces
- Each face represented independently
- If polygon soup, vertex count per face

Polygon Soup:

Polygons F

k_1: x_{11}, y_{11}, z_{11} ,..., x_{1k_1}, y_{1k_1}, z_{1k_1}

k_2: x_{21}, y_{21}, z_{21} ,..., x_{2k_1}, y_{2k_1}, z_{2k_1}

...

k_F: x_{F1}, y_{F1}, z_{F1} ,..., x_{Fk_F}, y_{Fk_F}, z_{Fk_F}

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex? (E-F+V=2-2g)

Triangle Soup

- List of faces
- Each face represented independently

Triangle Soup:

Triangles F

$x_{11}, y_{11}, z_{11}, x_{12}, y_{12}, z_{12}, x_{13}, y_{13}, z_{13}$

$x_{21}, y_{21}, z_{21}, x_{22}, y_{22}, z_{22}, x_{23}, y_{23}, z_{23}$

...

$x_{F1}, y_{F1}, z_{F1}, x_{F2}, y_{F2}, z_{F2}, x_{F3}, y_{F3}, z_{F3}$

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex?

Polygon/Triangle Soup

Advantage:

- General

Disadvantages:

- Redundant storage
- No (chance for) connectivity information
- [Polygon] variable-size data structures

Structure of Meshes

Common:

- Connected (small # of connected components)
- Two-manifolds
- Water-Tight (?)
- Orientable (?)
- Triangle meshes (?)

Indexed Polygon Set (Take 1)

- Lists of vertices, faces
- Each vertex stores geometry
- Each face points to its vertices

Vertex List:

Vertices V

x_{11}, y_{11}, z_{11}

...

x_{V1}, y_{V1}, z_{V1}

Polygon List:

Polygons F

k_1: v_{11} ,..., v_{1k_1}

...

k_F: v_{F1} ,..., v_{Fk_F}

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex?

Indexed Triangles Set (Take 1)

- Lists of vertices, faces
- Each vertex stores geometry
- Each face points to its vertices

Vertex List:

Vertices V

x_{11}, y_{11}, z_{11}

...

x_{V1}, y_{V1}, z_{V1}

Triangle List:

Triangles F

v_{11}, v_{12} , v_{13}

...

v_{F1} , v_{F2} , v_{F3}

Q: If the mesh is manifold with low genus, what is the storage cost, per vertex?

Indexed Triangles Set (Take 1)

- Lists of vertices, faces
- Each vertex stores geometry
- Each face points to its vertices

Vertex List:

Vertices V

x_{11}, y_{11}, z_{11}

...

x_{V1}, y_{V1}, z_{V1}

Triangle List:

Triangles F

v_{11}, v_{12} , v_{13}

...

v_{F1} , v_{F2} , v_{F3}

Q: If the mesh is completely disconnected, what is the storage cost, per vertex?

Indexed Polygon/Triangle Set (Take 1)

Advantage:

- Reduced storage

Disadvantages:

- Doesn't leverage manifold structure
- No (explicit) connectivity information
- Can't store per-edge information
- [Polygon] variable-size data structures

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes

Indexed Triangles Set (Take 2)

- Lists of vertices, faces
- Each vertex stores geometry + neighbor face
- Each face points to its vertices + neighbor faces

Vertex List:

Vertices V

$x_{11}, y_{11}, z_{11}; f_1$

...

$x_{V1}, y_{V1}, z_{V1}; f_V$

Triangle List:

Triangles F

$i_{11}, i_{12}, i_{13}; f_{11}, f_{12}, f_{13}$

...

$i_{F1}, i_{F2}, i_{F3}; f_{F1}, f_{F2}, f_{F3}$

Q: If the mesh is manifold with low genus, what is the storage cost, per vertex?

Indexed Triangles Set (Take 2)

Q: Given a vertex, how do we enumerate all of the vertices in its one-ring?

Indexed Polygon/Triangle Set (Take 2)

Advantages:

- Reduced storage
- Explicit connectivity information

Disadvantages:

- Partially leverages manifold structure
- Messy connectivity information
- Can't store per-edge information
- [Polygon] variable-size data structures

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes
Indexed P. Set (2)	Yes/No	Yes	Yes	Yes

Winged-Edge Mesh

- Lists of vertices, edges, faces
- Each vertex stores geometry + neighbor edge
- Each face stores neighbor edge
- Each edge points to vertices (x2), faces (x2), next edge (x2), and previous edge (x2)

Vertex List:

Vertices V

$x_{11}, y_{11}, z_{11}; e_1$

...

$x_{V1}, y_{V1}, z_{V1}; e_V$

Edge List:

Edges E

$v_{11}, v_{12}; f_{11}, f_{12}; n_{11}, n_{12}, p_{11}, p_{12}$

...

$v_{E1}, v_{E2}; f_{E1}, f_{E2}; n_{E1}, n_{E2}, p_{E1}, p_{E2}$

Face List:

Faces F

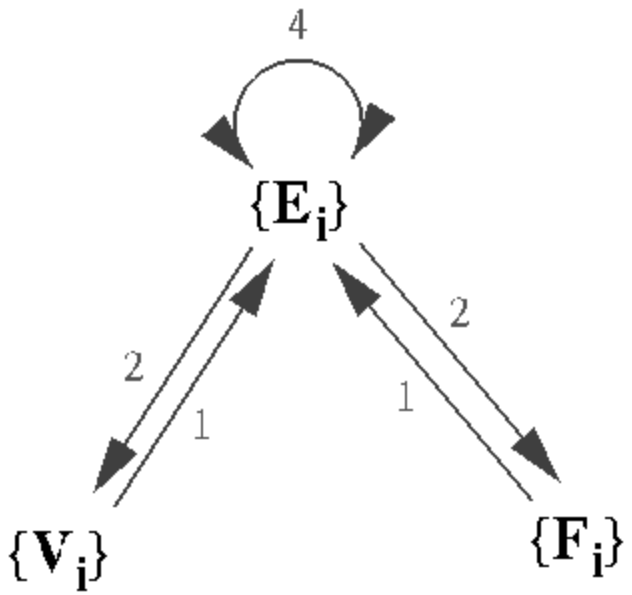
e_1

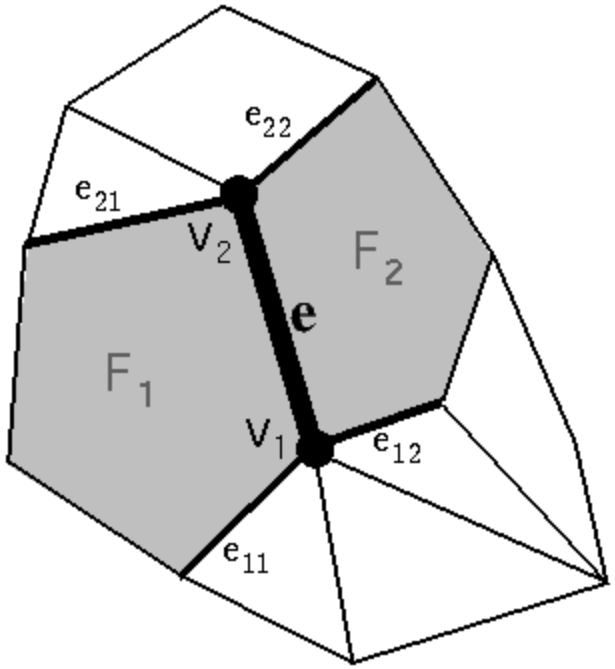
...

e_F

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex?

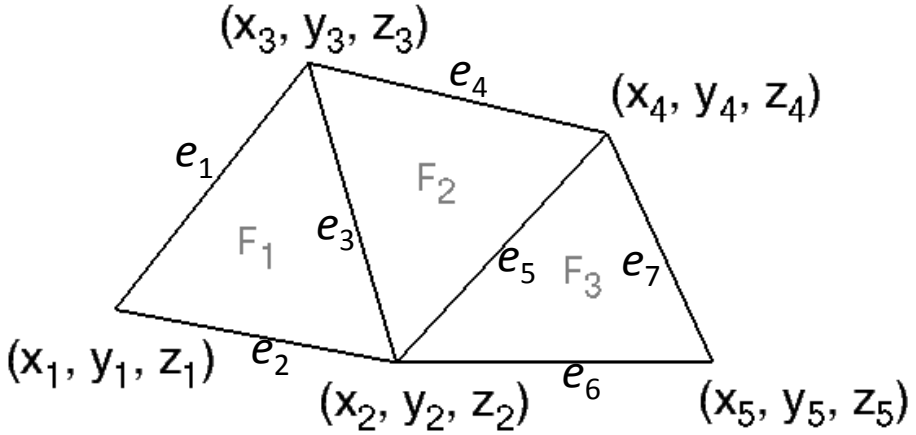
Winged-Edge Mesh





Winged-Edge Mesh

- Example:



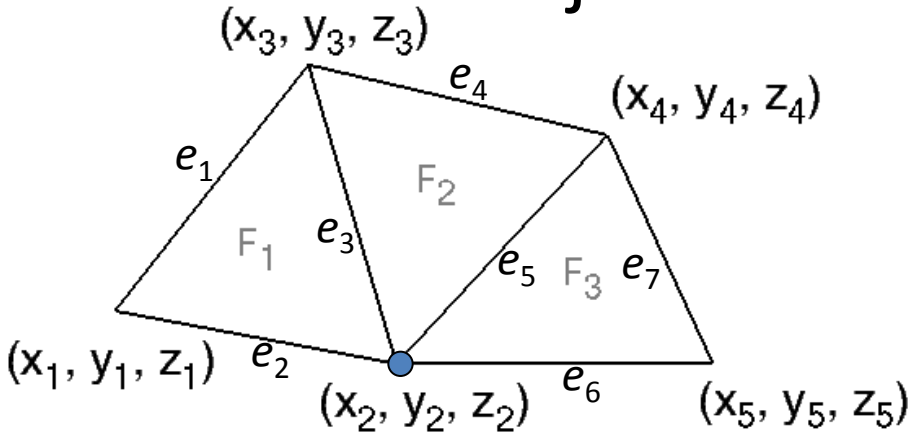
VERTEX TABLE				
V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE	
F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



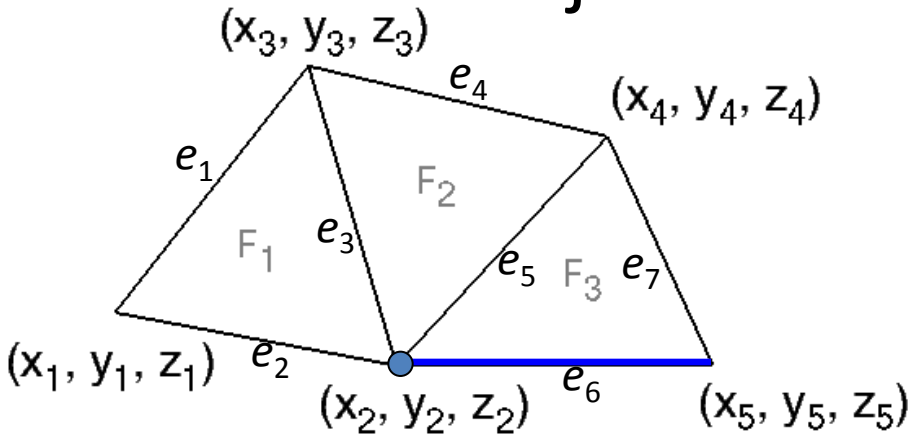
VERTEX TABLE				
V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE	
F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

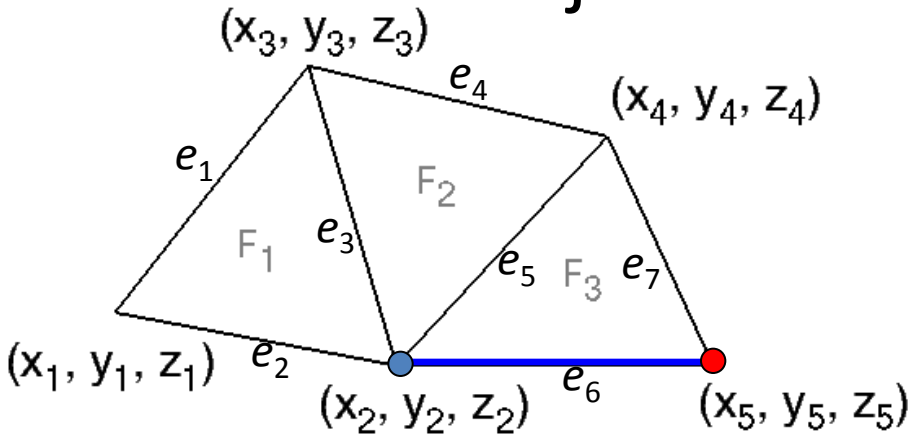
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

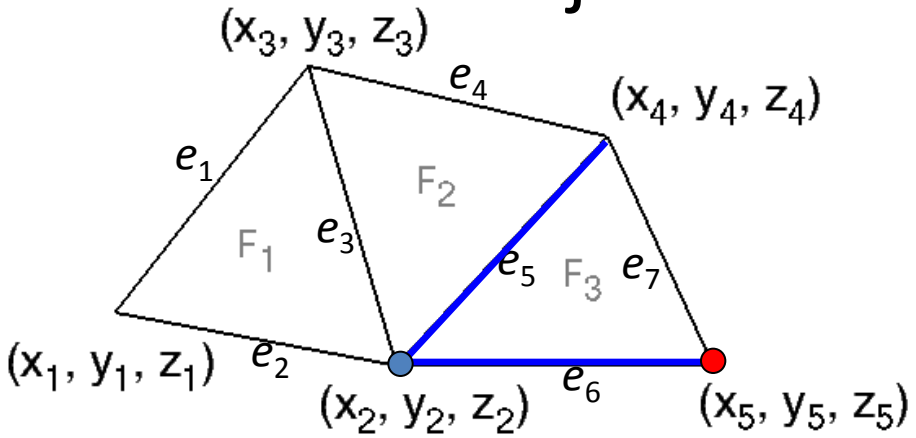
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

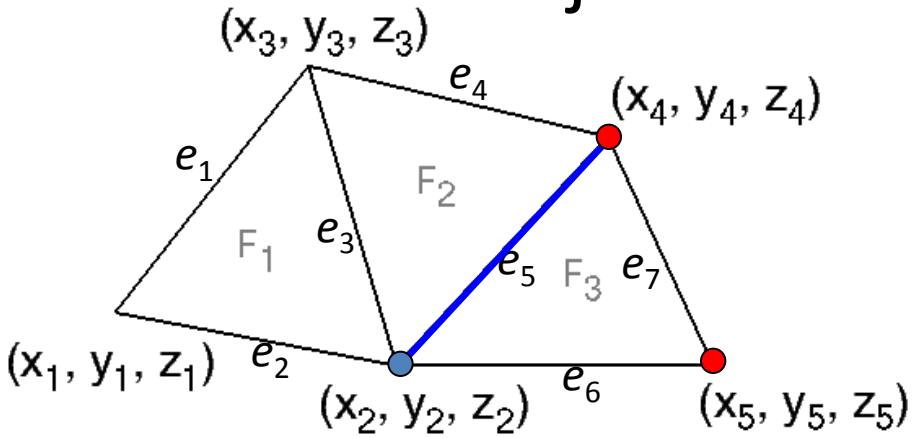
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

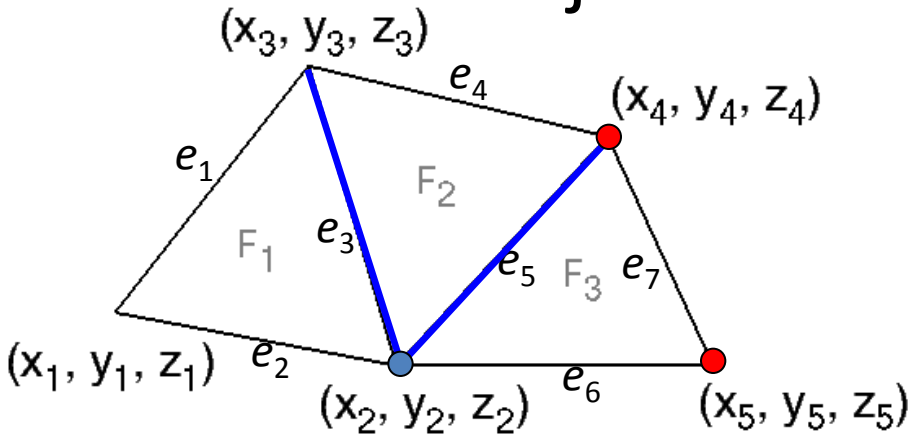
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V_3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

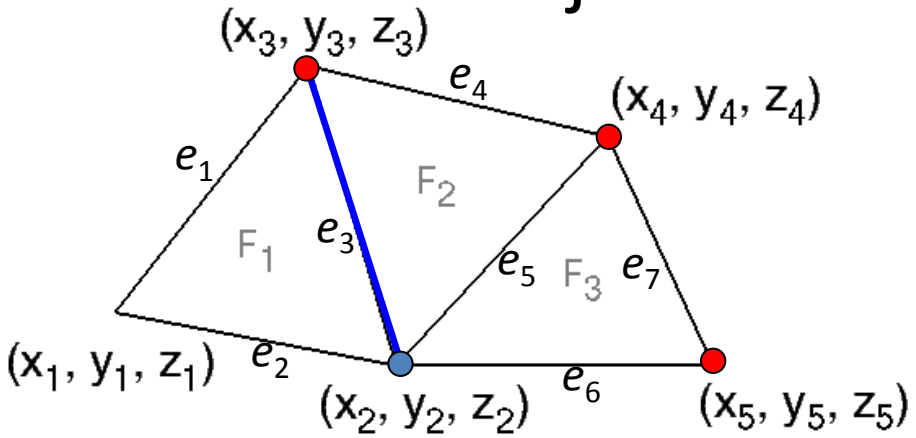
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V_3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

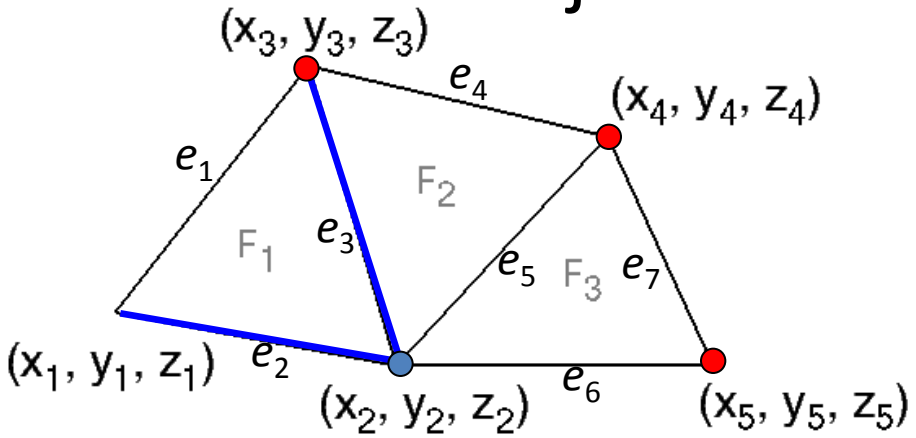
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V_3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

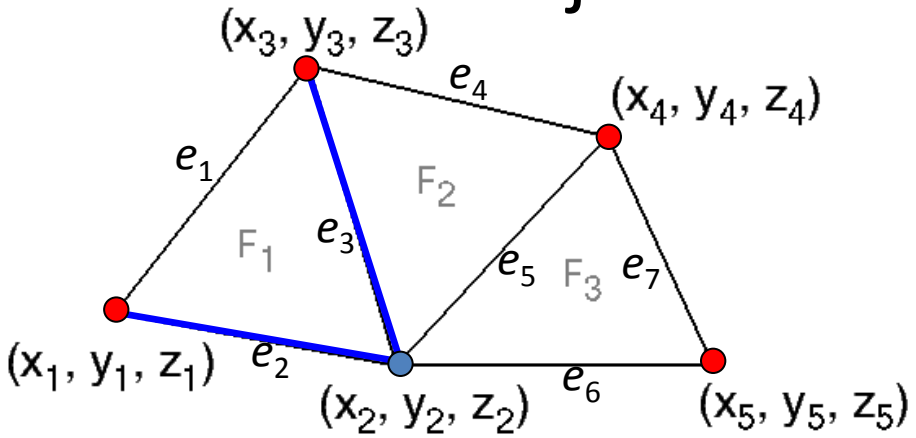
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V_3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

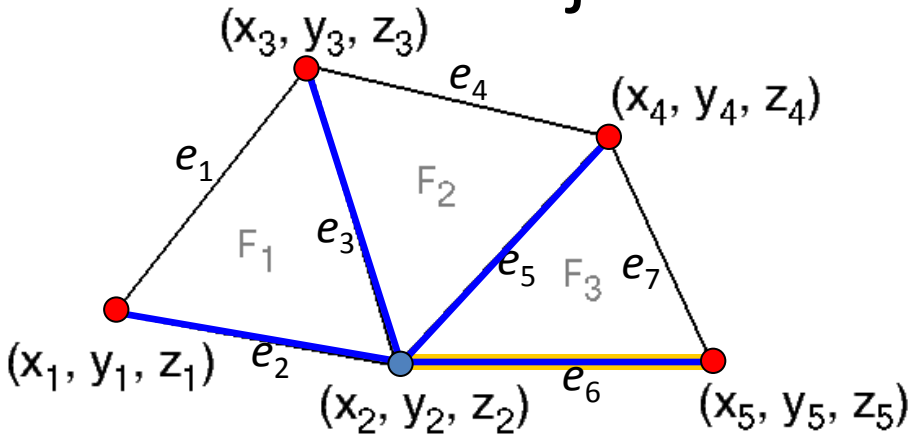
					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

- Example: Find vertices adjacent to v_2.



VERTEX TABLE

V_1	X_1	Y_1	Z_1	e_1
V_2	X_2	Y_2	Z_2	e_6
V_3	X_3	Y_3	Z_3	e_3
V_4	X_4	Y_4	Z_4	e_5
V_5	X_5	Y_5	Z_5	e_6

EDGE TABLE

					11	12	21	22
e_1	V_1	V_3		F_1	e_2	e_2	e_4	e_3
e_2	V_1	V_2	F_1		e_1	e_1	e_3	e_6
e_3	V_2	V_3	F_1	F_2	e_2	e_5	e_1	e_4
e_4	V3	V_4		F_2	e_1	e_3	e_7	e_5
e_5	V_2	V_4	F_2	F_3	e_3	e_6	e_4	e_7
e_6	V_2	V_5	F_3		e_5	e_2	e_7	e_7
e_7	V_4	V_5		F_3	e_4	e_5	e_6	e_6

FACE TABLE

F_1	e_1
F_2	e_3
F_3	e_5

Winged-Edge Mesh

Advantages:

- Reduced storage
- Explicit connectivity information
- Can store per-edge information
- Fixed-size data-structures

Disadvantage:

- Messy connectivity information

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes
Indexed P. Set (2)	Yes/No	Yes	Yes	Yes
Winged-Edge	No	Yes	Yes	Yes

Half-Edge Mesh (Take 1)

- Lists of vertices, half-edges, faces
- Each vertex stores geometry + half-edge
- Each face stores neighbor half-edge
- Each half-edge points to start vertex, face, opposite/next/previous half-edge

Vertex List:

Vertices V

$x_{11}, y_{11}, z_{11}; e_1$

...

$x_{V1}, y_{V1}, z_{V1}; e_V$

Half-Edge List:

Edges E

$v_1; f_1; o_1, n_1, p_1$

...

$v_E; f_E; o_E, n_E, p_E$

Face List:

Faces F

e_1

...

e_F

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex?

Half-Edge Mesh (Take 1)

Q: Given a vertex, how do we enumerate all of the vertices in its one-ring?

Half-Edge Mesh (Take 1)

Advantages:

- Reduced storage
- Connectivity information is easy
- Fixed-size data structures

Disadvantage:

- Cannot store per-edge information

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes
Indexed P. Set (2)	Yes/No	Yes	Yes	Yes
Winged-Edge	No	Yes	Yes	Yes
Half-Edge (1)	No	No	Yes	Yes

Half-Edge Mesh (Take 2)

- Lists of vertices, pairs of half-edges, faces
- Each vertex stores geometry + half-edge
- Each face stores neighbor half-edge
- Each half-edge points to start vertex, face, and next half-edge

Vertex List:	**Half-Edge List**:	**Face List**:
Vertices V	Edges E	Faces F
x_{11}, y_{11}, z_{11}; e_1	v_1; f_1; n_1	e_1
...	...	...
x_{V1}, y_{V1}, z_{V1}; e_V	v_E; f_E; n_E	e_F

Q: If the mesh is manifold, triangular with low genus, what is the storage cost, per vertex?

Half-Edge Mesh (Take 2)

Q: Given a vertex, how do we enumerate all of the vertices in its one-ring?

Half-Edge Mesh (Take 2)

Advantage:

- Reduced storage
- Connectivity information is easy
- Can store per-edge information
- Fixed-size data structures

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes
Indexed P. Set (2)	Yes/No	Yes	Yes	Yes
Winged-Edge	No	Yes	Yes	Yes
Half-Edge (1)	No	No	Yes	Yes
Half-Edge (2)	No	No	Yes	Yes

Directed-Edge Triangle Mesh

- Lists of vertices, half-edges
- Each vertex stores geometry + half-edge
- Each half-edge points to start vertex and opposite half-edge

Vertex List:

Vertices V

$x_{11}, y_{11}, z_{11}; e_1$

...

$x_{V1}, y_{V1}, z_{V1}; e_V$

Half-Edge List:

Edges E

$v_1; o_1$

...

$v_E; o_E$

Q: If the mesh is manifold with low genus, what is the storage cost, per vertex?

Directed-Edge Triangle Mesh

Q: Given a vertex, how do we enumerate all of the vertices in its one-ring?

Directed-Edge Triangle Mesh

Advantages:

- Reduced storage
- Connectivity information is easy

Disadvantage:

- Cannot store per-edge information

	Non-Manifold	Non-Orientable	General Polygons	Boundaries
Indexed P. Set (1)	Yes	Yes	Yes	Yes
Indexed P. Set (2)	Yes/No	Yes	Yes	Yes
Winged-Edge	No	Yes	Yes	Yes
Half-Edge (1)	No	No	Yes	Yes
Half-Edge (2)	No	No	Yes	Yes
Directed-Edge	No	Yes	No	No