Edge Vertex relationships and EdgeBreaker optimization

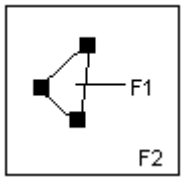
Polygons are:

- Planar Graphs
- Connected. This means:
  - No holes
  - No faces with holes

V – Vertices
E – Edges
F – Faces

Polygon relationships:

- V = E
- V-E=0
- F = 2



Then using induction:
F = F + 1
V = V + 2 and (V – E) + F = 2 since [(V+2) – (E + 3)] + F + 1 = 2.
E = E + 3.

Primitive Operations

Adding a vertex preserves V – E + F = 2
Adding an edge also preserves V – E + F = 2

Valence

- The valence of a vertex V is the number of incident edges.
- The valence of face F is the number of bounding vertices.
- For a t-mesh the average vertex valence is approximately 6

Characterizing a mesh in terms of vertex valence K allows identification of meshes in terms of the primitives used to build the mesh.

So transforming V-E+F = 2 to an equation using (V, F, K):

K = 3. Edge uses = 6.
So Sum K = 2E.

Edges are used by vertices KV.
Each edge is used twice 2E.

2E = KV
E = (K/2)V

V – (K/2)V + F = 2
(1 – (K/2))V + F = 2
((2 – (K/2))/2)V + F = 2.

If a T-Mesh is assumed, then:

Edge uses is 3T, and each edge is used twice 2E so:

2E = 3T
E = (3/2)T
V-(3/2)T + T = 2
2V-T=4
T=2V-4

Improving Edge Breaker:

- Certain combinations of letters are not possible. This can be used as an optimization.
- Letters with offset encoding can be used to replace runs.
- S triangles can be reduced.
- Geometry can be used to predict encoding