**Edgebreaker: Connectivity compression for triangle mesh**

Jarek Rossignac

Steven Chang
21/02/2006
CPSC533A-Digital Geometry Processing
Paper presentation

**Edgebreaker**

- A simple scheme for compressing the triangle-vertex incidence graphs of 3D triangle meshes.
- Connectivity compression
  - Less than 2 bits per triangle

**Outline**

- The motivation
- Brief review of prior art
- Edgebreaker algorithm
- General triangle meshes
- Question

**The motivation**

- 3D models are usually large
  - Triangle mesh is often stored as two tables, one for triangle-vertex incidence (connectivity), one for coordinates (geometry).
  - When pointers or integer indices are used as vertex-references and when floating point coordinates are used to encode vertex locations, uncompressed connectivity data consumes twice more storage than vertex coordinates.
- Bandwidth:
  - It is all about time.
- Storage:
  - Resources and money

**Brief review of prior art**

- Uncompressed data structures
- Triangle strips
- Progressive vertex insertion
- Graph encoding
- Vertex permutation

**Edgebreaker algorithm**

- A guaranteed, low, linear storage cost compression for simple mesh, namely, the triangle mesh that forms a connected, orientable, manifold surface that is homeomorphic to a sphere or to a half-sphere.
- Edgebreaker organizes the vertices of the model along a spiraling vertex-spanning tree
- Edgebreaker uses a stack to keep track of every loop.
**Terminology**
- Interior edge, exterior edge
- Interior vertices, exterior vertices
- Region, boundary, gate
- Stack

**Cases**
- 5 cases, C, L, E, R, S

**Example**

```java
if (v not in B) {
    case C
} else if (v follows g) {
    if (v precedes g) case E
    else case R
} else {
    if (v precedes g) case L
    else case S
}
```

**Example**

```
H=CCRRRSLCRSERELCRRRCRRRE
```

**Encoding**
- We have to encode a string comprising of 5 distinct letters
- Notice that there is one-to-one association between vertices and the triangles processed by an C operation
- One possible set:

<table>
<thead>
<tr>
<th>C</th>
<th>L</th>
<th>E</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>110</td>
<td>111</td>
<td>101</td>
<td>100</td>
</tr>
</tbody>
</table>

**Encoding (cont.)**
- Total number of bits:
  \[ c = |C| + 3(|S| + |L| + |R| + |E|) \]
- Euler says for simple meshes:
  \[ |T| = 2|V| \]
- Hence, connectivity cost:
  \[ c \sim 2|T| \]
  i.e. 2 bits/triangle
**Encoding (cont.)**
- Two impossible combinations: CL, CE
- Modify the previous coding a little bit. Code set for operations do not follow an C:
  
<table>
<thead>
<tr>
<th>C</th>
<th>L</th>
<th>E</th>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>110</td>
<td>111</td>
<td>101</td>
<td>100</td>
</tr>
</tbody>
</table>
- Code set for operations follow an C
  
<table>
<thead>
<tr>
<th>CC</th>
<th>CR</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>
- 1.5 bits/triangle

**Decompression**

Two phases: **Preprocessing** and **Generation**

**Preprocessing phase:** Edgebreaker reads the input stream, i.e.: an encoding of the sequence of op-codes, decodes the op-code one at a time, stores them in H for generation phase.

  - Total number of operations = triangle count
  - Process finishes when |S|-|E|<0
  - At the end of phase, |T|=t, |Vi|=c, |Ve|=e and O contains offsets.
  - External edges: e=3|E|+|L|+|R|-|C|-|S|
  - Offset is the difference of e, the number of edges/vertices, between corresponding S and E, and used to get the third vertex of the triangle with S operation during generation phase

**Decompression (cont.)**

**Generation phase:** Edgebreaker allocates a table of triangle-vertex incidence relations (connectivity), denoted TV, of |T| entries, each will combine the three integer vertex labels of a triangle

- Each operation creates a new triangle adjacent to active gate.

**General triangle meshes**

- The Edgebreaker approach is capable of encoding the connectivity of any planar triangle graph with zero or one hole.

- We want to extend the Edgebreaker compressed format and the compression algorithms, so as to support meshes with multiple holes and with one or more handles.

**General triangle meshes (cont.)**

- Holes: more than one bounding loop
- Add a new case: M, i.e., during an S operation, we merge loops rather than splitting the current loop into two.
General triangle meshes (cont.)

- Need to further modify the encoding strategy as follows:
  
<table>
<thead>
<tr>
<th>C</th>
<th>L</th>
<th>E</th>
<th>R</th>
<th>S</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>110</td>
<td>111</td>
<td>101</td>
<td>1000</td>
<td>1001</td>
</tr>
</tbody>
</table>

- Add \(|S|\) bits to the overall compressed representation in addition to the cost of encoding the \(M\) operation

General triangle meshes (cont.)

- Handles: \(S\) operation creates holes


General triangle meshes (cont.)

- So modify the \(S\) operation

- Add another \(M'\) operation

Questions?

Reference:

- CPSC533A Lecture notes, “Compressing Connectivity”
- Davis King, 3D compression and shape complexity GVU center and college of computing, Georgia Tech.