Higher-dimensional object modelling in GIS based on G-maps

Ken Arroyo Ohori
G.A.K.ArroyoOhori@tudelft.nl
Part of a 5D project

• Full integration of 2D/3D space, time and scale dimensions
  • Treating these as spatial (actual geometry)

• A 5D continuum

• Managing and querying data more efficiently

• Ensure consistency across all dimensions
What I aim to do

• Realise data models, data structures and operations for higher dimensional ( > 3 ) spatial data.

  • Visualisation: how to display data in > 3D?

  • Operations: slicing and projections

• Take advantage of research done in other fields. Apply it to GIS data, which is “special”.
an example: G-maps and GIS data
$$\alpha_0$$

$$\alpha_1$$

$$\alpha_2$$

$$\phi_2 = \langle \alpha_0, \alpha_1 \rangle$$

$$\phi_1 = \langle \alpha_0, \alpha_2 \rangle$$

$$\phi_0 = \langle \alpha_1, \alpha_2 \rangle$$

G-maps

2D
G-maps
G-maps

• Data model

• Combinatorial (topological) structure

• How to store geometry?

• How to store attributes?

```c
struct Dart {
    Dart *involutions[n+1];
};
```

```c
struct Involution {
    id dart1, dart2;
};
```
Why GIS data is “special”

- Storing geometry, topology and computed values
- Complex handling of attributes: numeric, text, classes
- Construction from invalid or non topological data
- Queries: geometric, topological, attribute based, or a combination
- Holes, in possibly every dimension > 0
- Disconnected objects
involutions\[n+1\]

Dart

darts[]
efficient access to all darts

ReferenceDart
topological operations

n-G-map

keep track of disconnected embeddings and provide high level operations

embeddings\[n+1\]
geometric operations

Embedding

holes[]
maintain a connected graph

store attributes, including geometry and the dimension of this embedding

embeddings[]
efficient access to all embeddings, e.g. to know if a certain embedding exists or not

What is needed then?
What is needed then?
### Memory consumption

<table>
<thead>
<tr>
<th>$n$</th>
<th>object</th>
<th>darts</th>
<th>embeddings</th>
<th>pointers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>point</td>
<td>0</td>
<td>$&lt;1&gt;= 1$</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>line segment</td>
<td>2</td>
<td>$&lt;2,1&gt;= 3$</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>triangle</td>
<td>6</td>
<td>$&lt;3,3,1&gt;= 7$</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>tetrahedron</td>
<td>24</td>
<td>$&lt;4,6,4,1&gt;= 15$</td>
<td>222</td>
</tr>
<tr>
<td>$n$</td>
<td>$n$-simplex</td>
<td>$\prod_{i=1}^{n}(i + 1)$</td>
<td>$2^{n+1} - 1$</td>
<td>$2e + 2d(n + 1)$</td>
</tr>
<tr>
<td>2</td>
<td>Figure 1</td>
<td>18</td>
<td>$&lt;5,7,3&gt;= 15$</td>
<td>138</td>
</tr>
<tr>
<td>3</td>
<td>Figure 2c</td>
<td>96</td>
<td>$&lt;12,20,11,2&gt;= 45$</td>
<td>858</td>
</tr>
<tr>
<td>3</td>
<td>Figure 3</td>
<td>128</td>
<td>$&lt;9,20,12,4&gt;= 45$</td>
<td>1114</td>
</tr>
</tbody>
</table>
several problems to solve...
(with any structure)
Difficulties to solve (efficiently)

• Construction (recreation of topology, comparison of two objects)

• Marking and storing temporary values

• Consistency and validity (geometry, topology, both)

• Keeping track of embeddings (e.g. deletion)

• Memory consumption
Other options

- Simplex-based
  - Problems: subdivision into simplices, redundancy

- Nef polyhedra based
  - Problems: recursive (and difficult) implementation
Future work

• Explore other options for data models and data structures

• Look into operations (e.g. slicing)

• Implement a visualiser for higher dimensional data

• Database implementation
Questions?