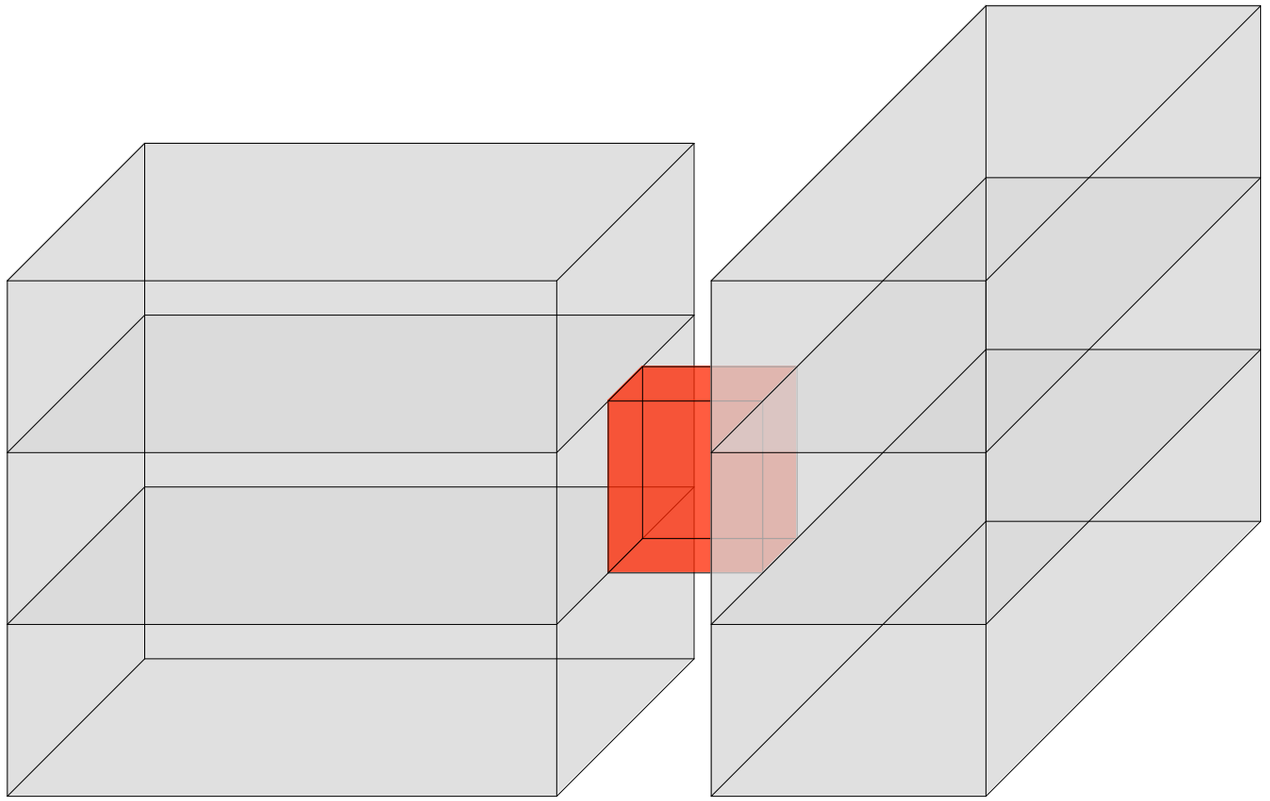
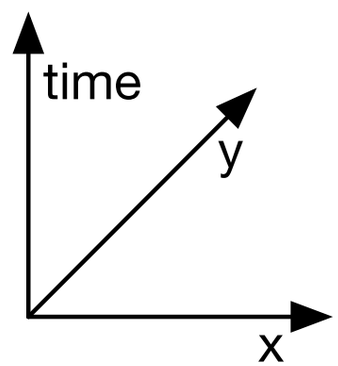
Higher-dimensional object modelling in GIS based on G-maps

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

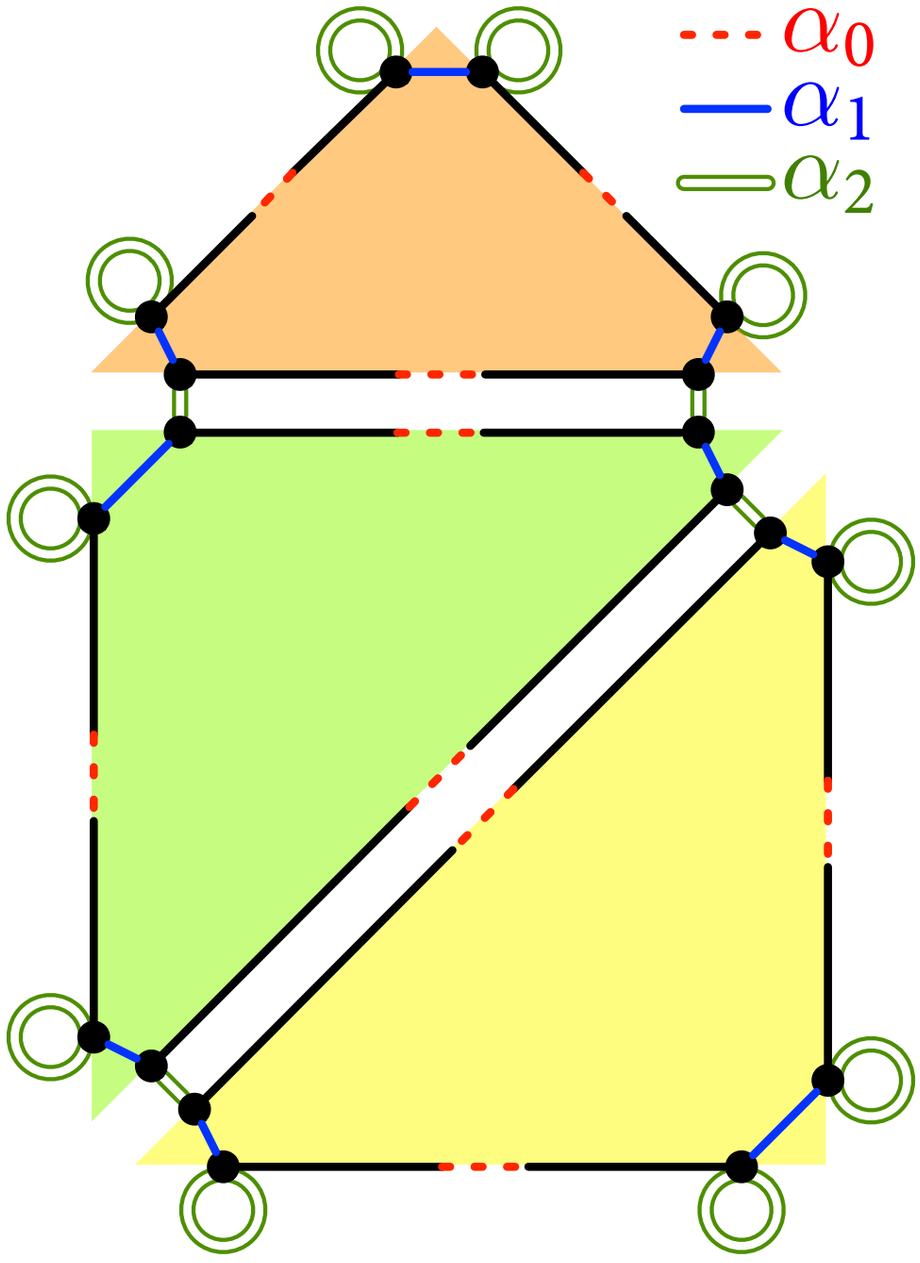


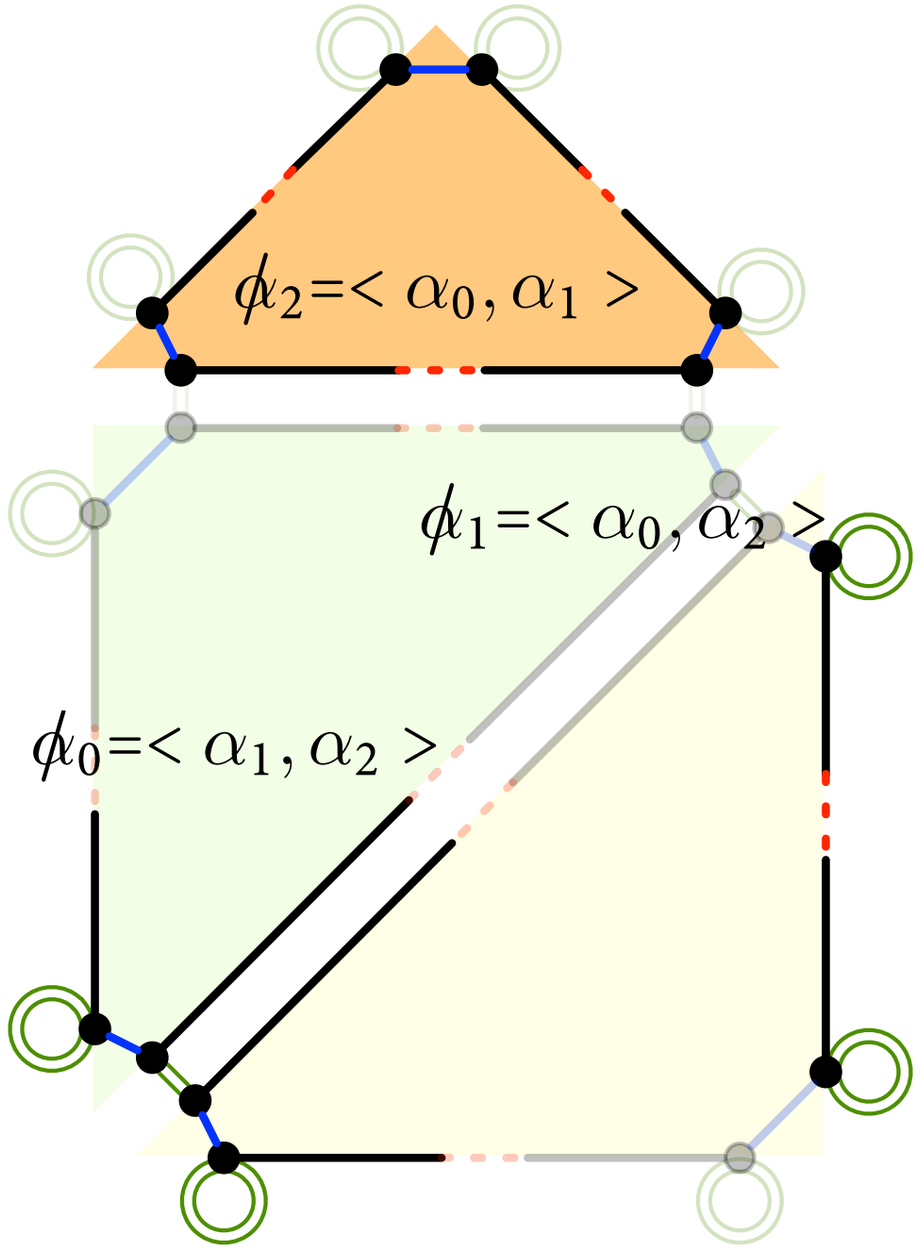


What I aim to do

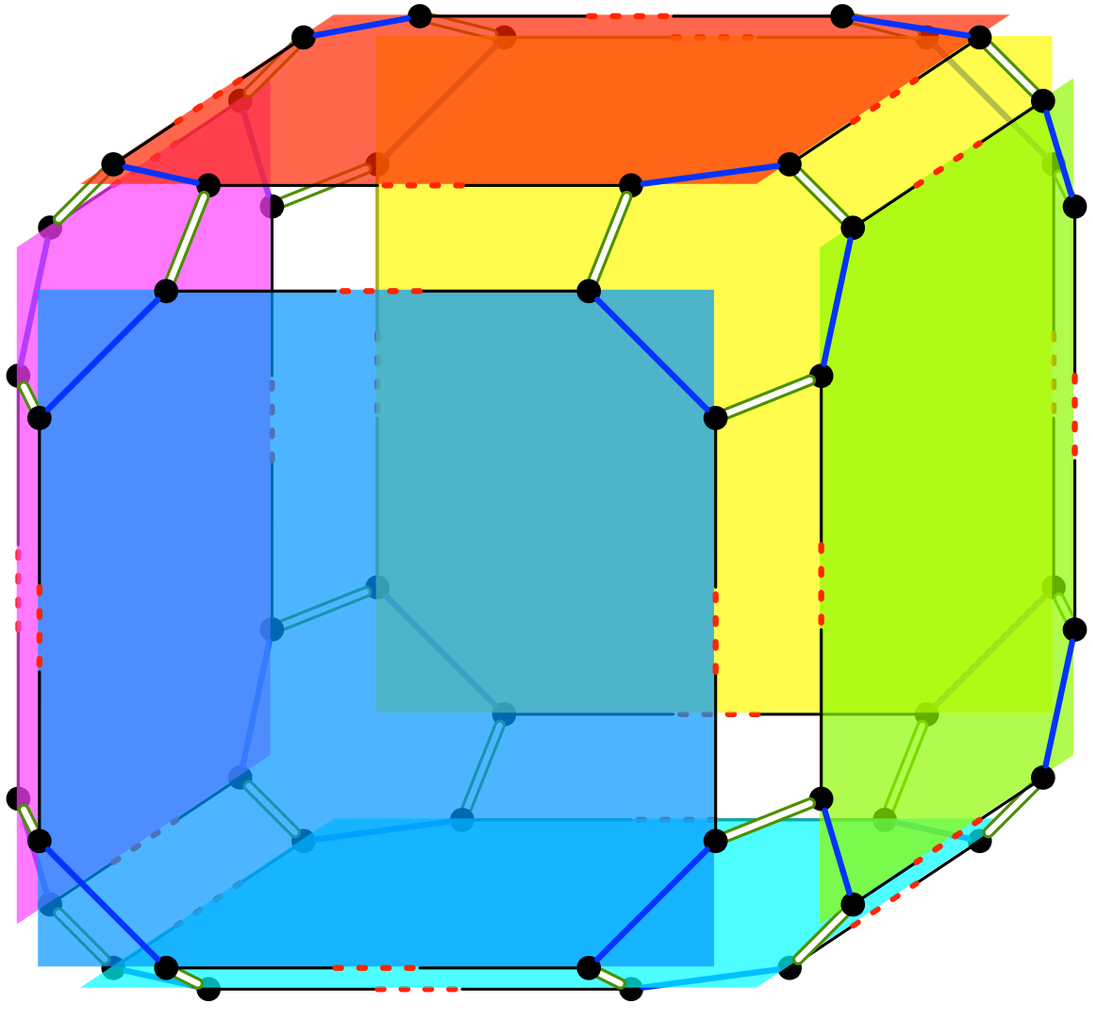
- Realise data models, data structures and operations for higher dimensional (> 3) spatial data.
    - Visualisation: how to display data in > 3D?
    - Operations: slicing and projections
- Take advantage of research done in other fields. Apply it to GIS data, which is “special”.

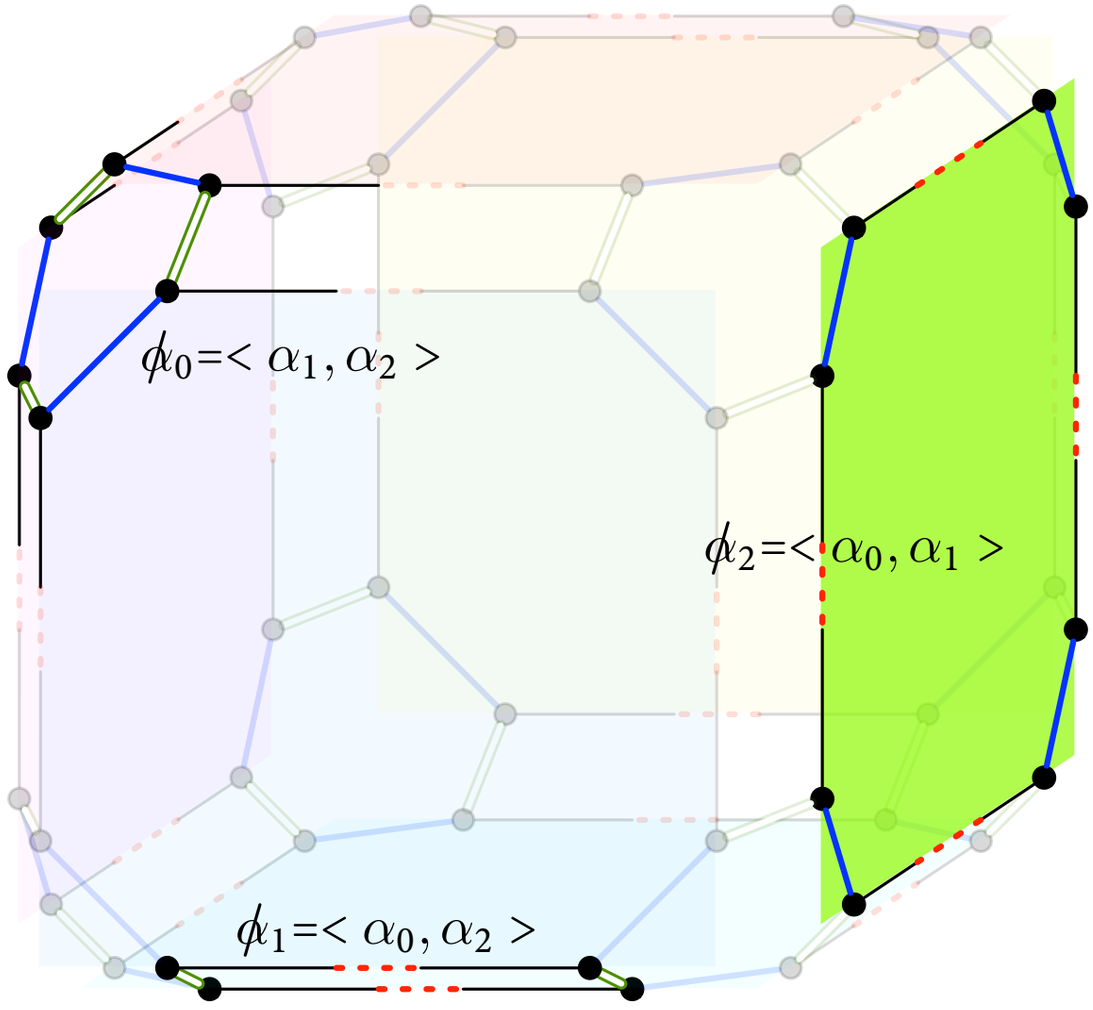
an example: G-maps and GIS data


α_0
α_1
α_2

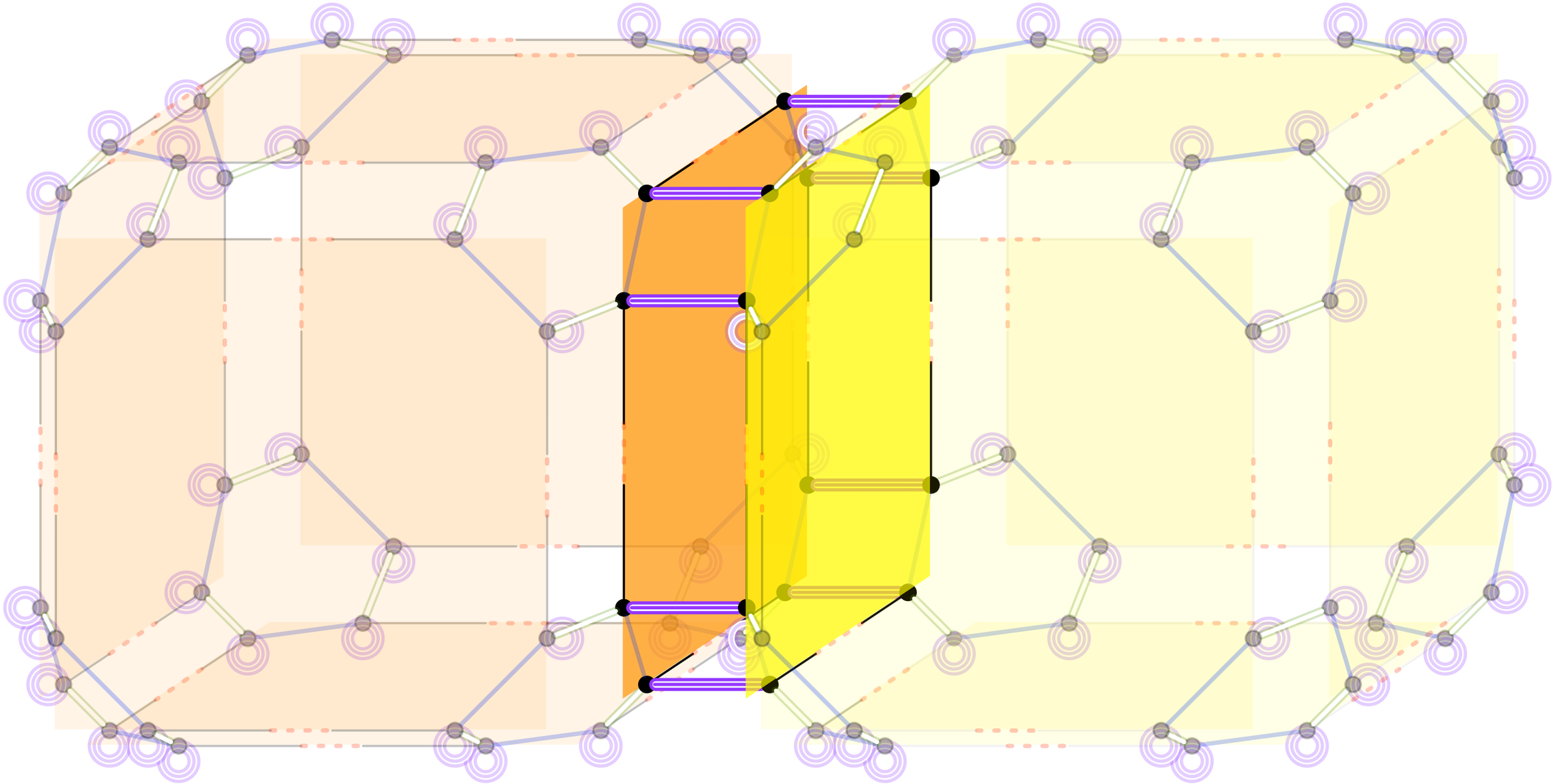

$\phi_2 = < \alpha_0, \alpha_1 >$
$\phi_1 = < \alpha_0, \alpha_2 >$
$\phi_0 = < \alpha_1, \alpha_2 >$

G-maps | 2D





G-maps | 3D



G-maps

3D

G-maps

- Data model

- Combinatorial (topological) structure

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

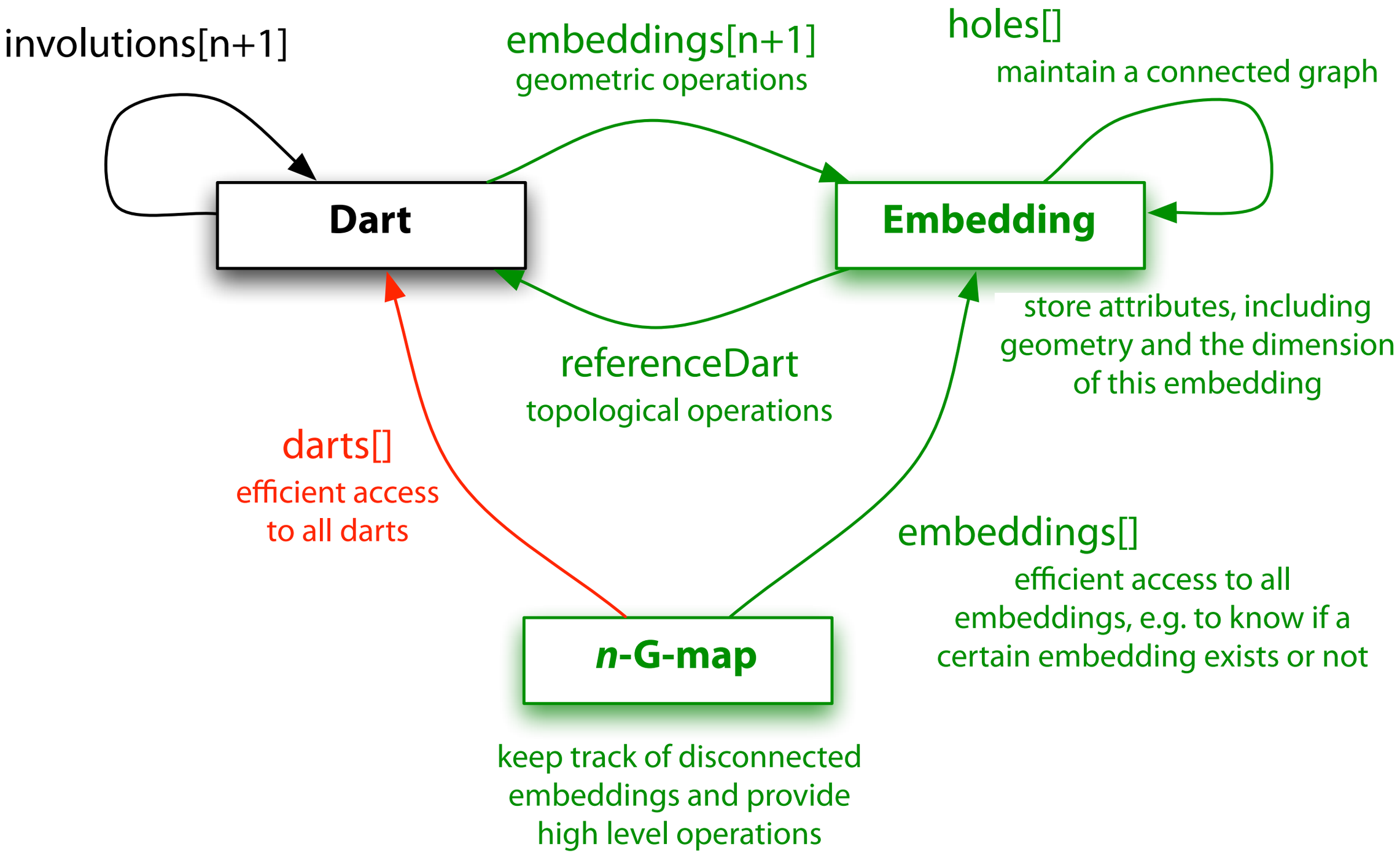
- How to store geometry?

- How to store attributes?

```
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

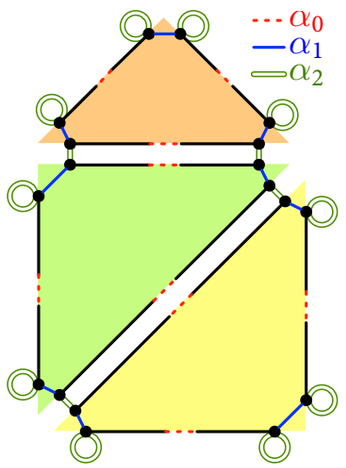


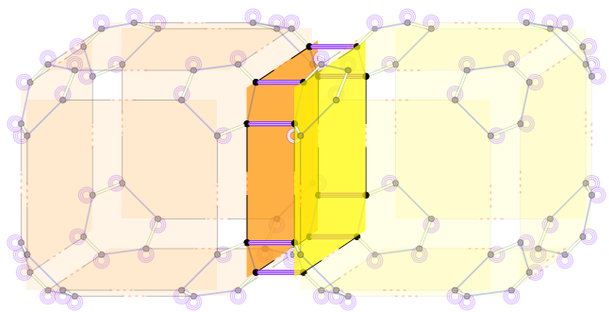
What is needed then?

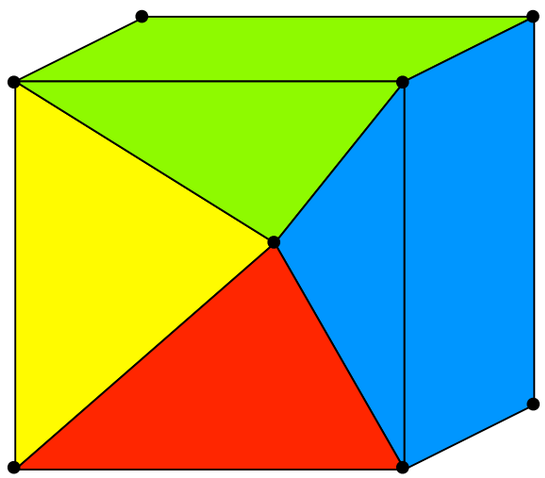
```
struct Dart {
    Dart *involutions[n+1];
    Embeddings *embeddings[n+1];
};

struct Embedding {
    Dart *referenceDart;
    Embedding *holes[];
    int dimension;
    ...
    float red, green, blue;
};

struct PointEmbedding : Embedding {
    float x, y, z;
};

struct GMap {
    Embedding* embeddings[];
};
```

What is needed then?

n	object	darts	embeddings	pointers
0	point	0	<1>= 1	2
1	line segment	2	<2,1>= 3	14
2	triangle	6	<3,3,1>= 7	50
3	tetrahedron	24	<4,6,4,1>= 15	222
n	n-simplex	$\prod_{i=1}^{n}(i+1)$	$2^{n+1}-1$	$2e+2d(n+1)$
2	Figure 1	18	<5,7,3>= 15	138
3	Figure 2c	96	<12,20,11,2>= 45	858
3	Figure 3	128	<9,20,12,4>= 45	1114







several problems to solve...

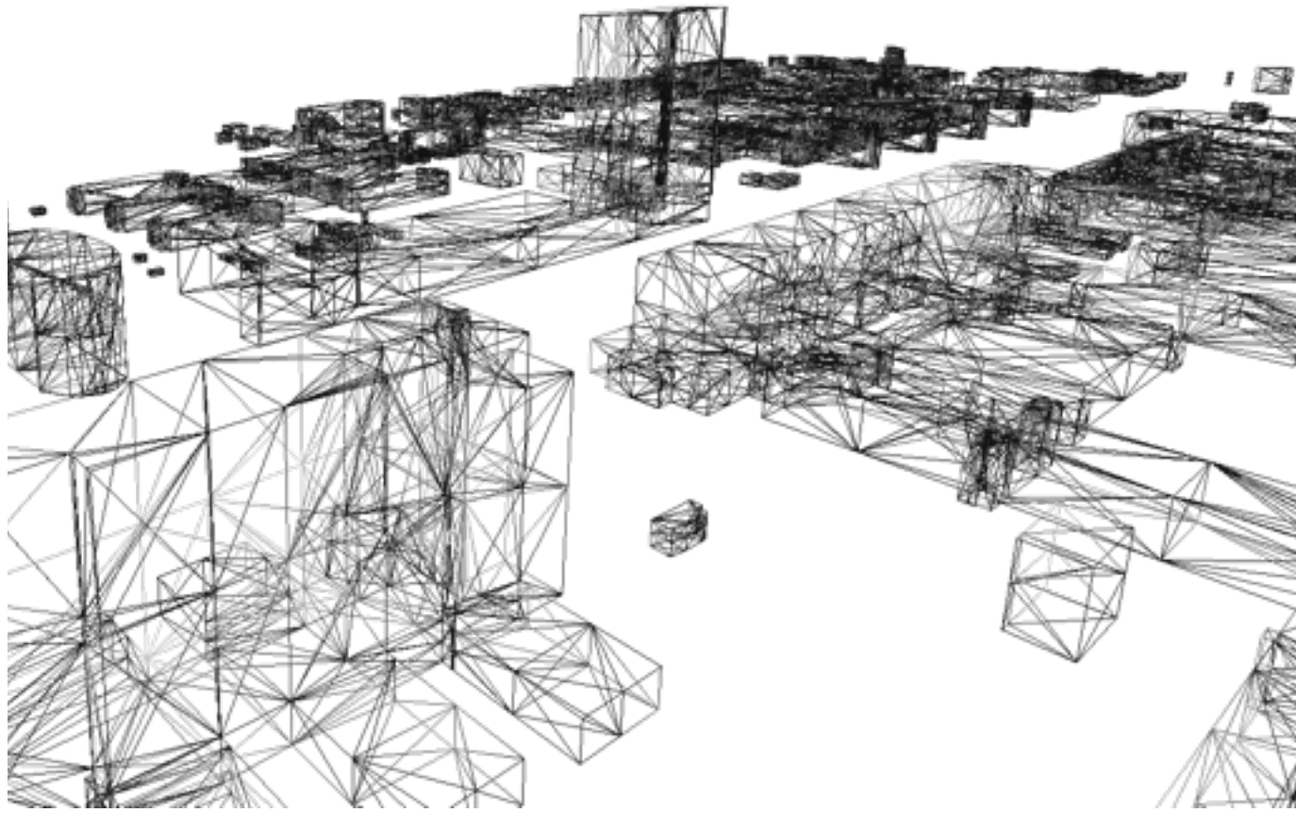
(with any structure)

Difficulties to solve (efficiently)

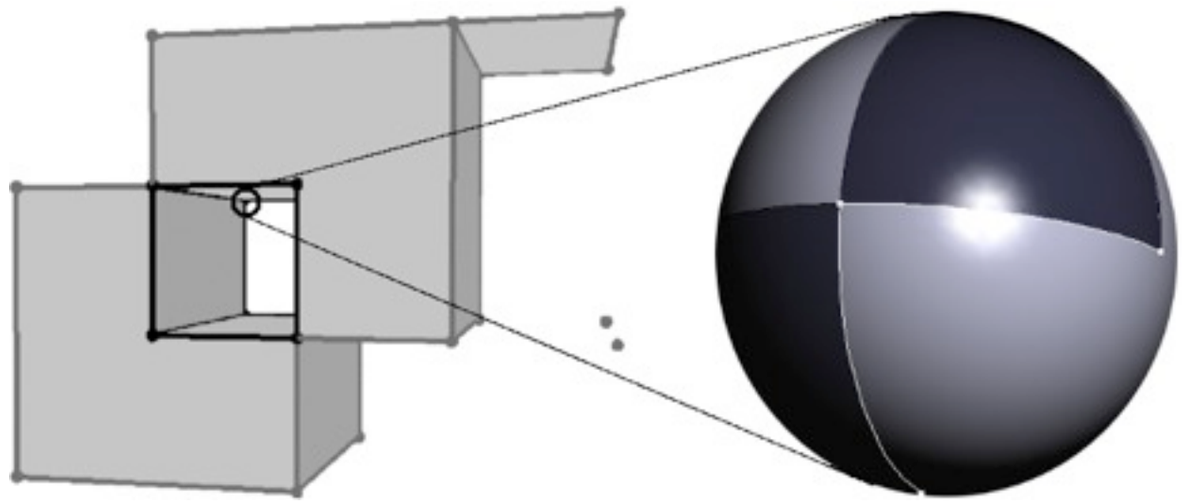
- Construction (recreation of topology, comparison of two objects)
- Marking and storing temporary values
- Consistency and validity (geometry, topology, both)
- Keeping track of embeddings (e.g. deletion)
- Memory comsuption

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?