Integrating scale and space in 3D city models

Jantien Stoter
Hugo Ledoux
Martijn Meijers
Ken Arroyo Ohori

TU Delft Delft University of Technology

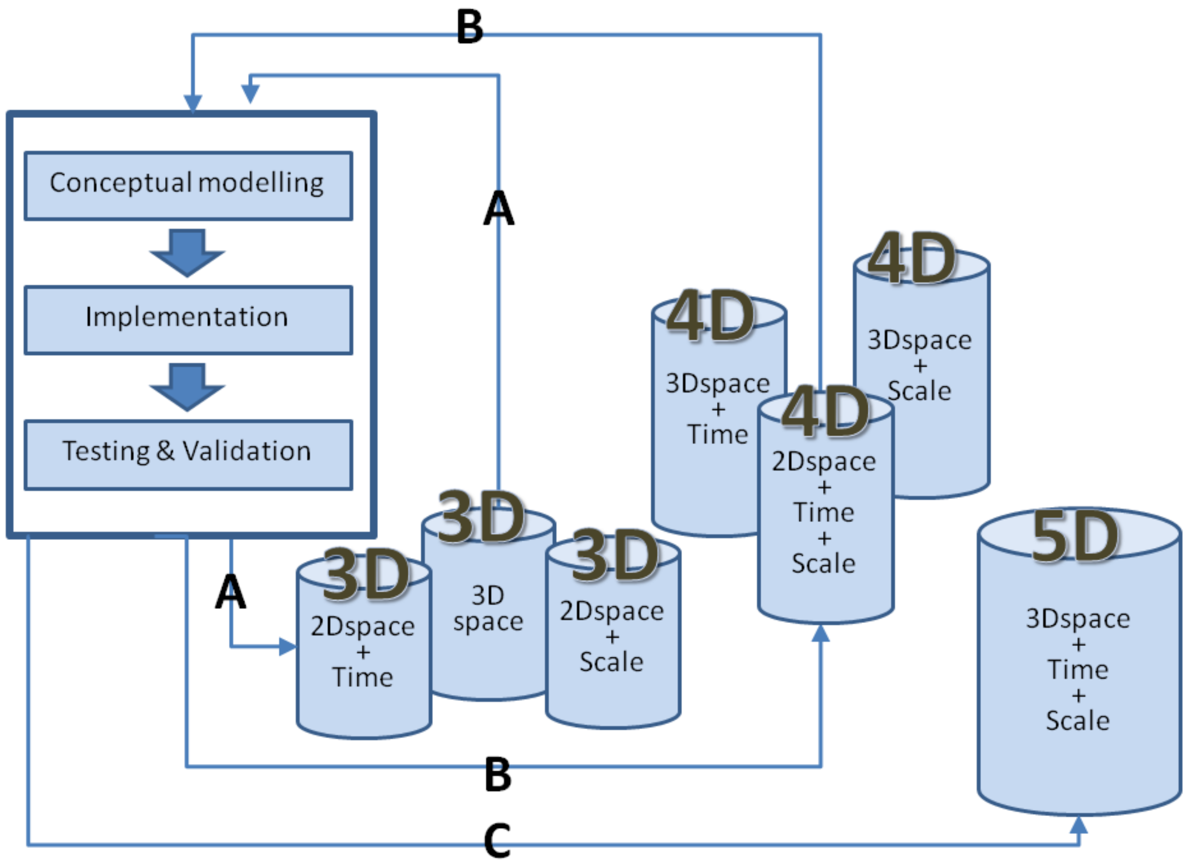
5D modelling project

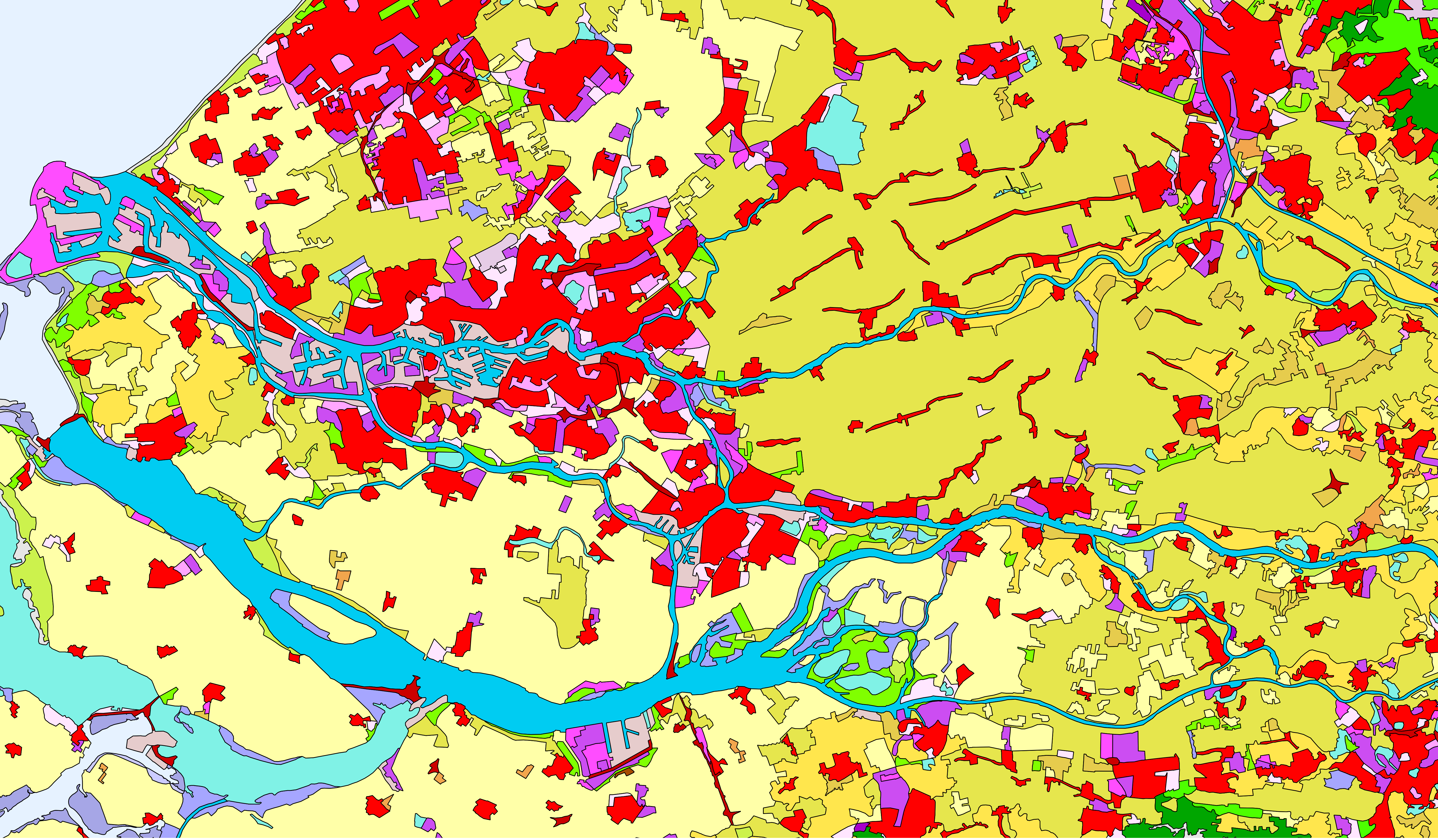
- Full integration of 3D space, time and scale dimensions

- Into a 5D hypercube

- Goal is to *implement* one

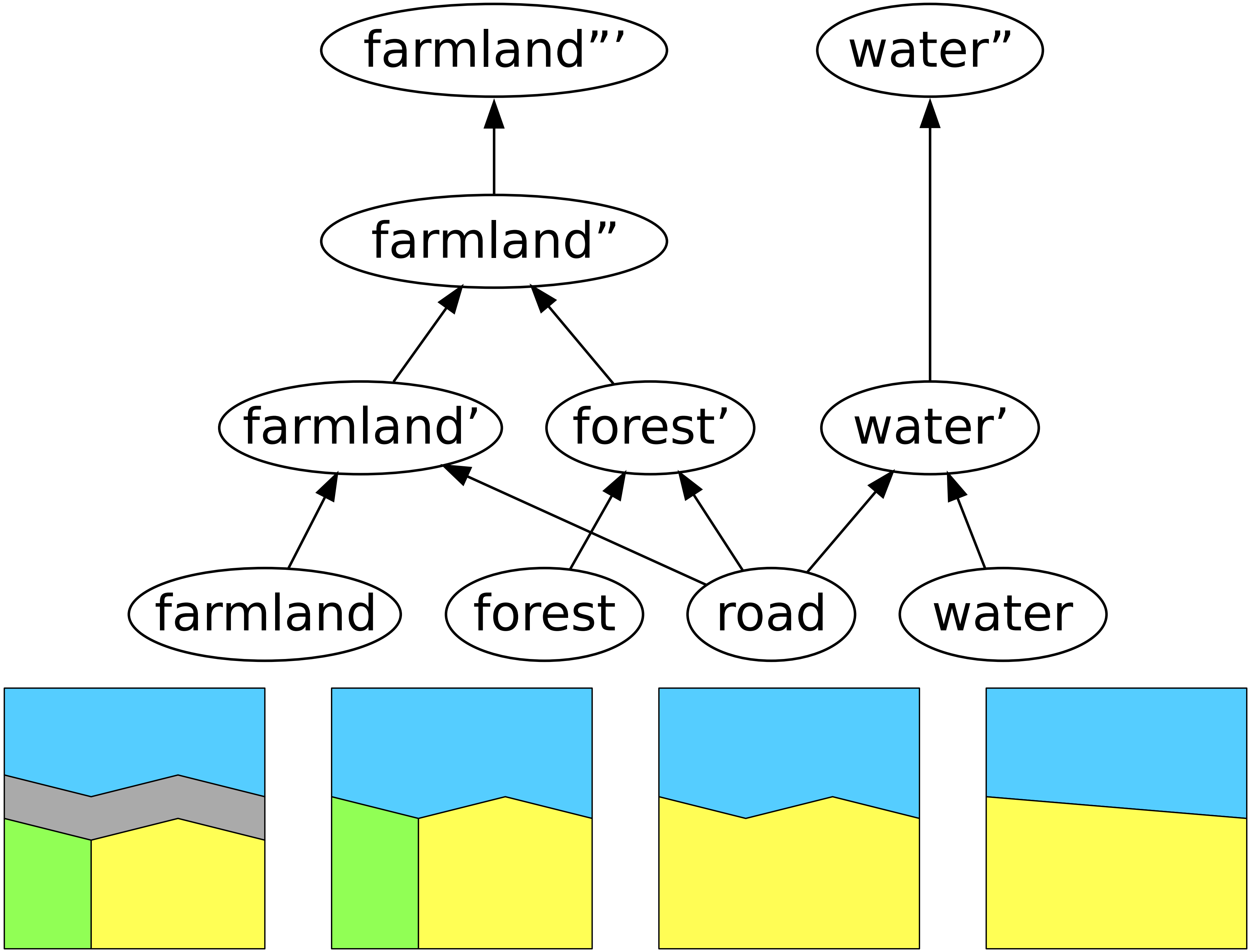
- Advantages:
  1. Ensures consistency across all dimensions (eg no gaps in space/ time)
  2. Querying across diff spaces
  3. Managing data more efficient





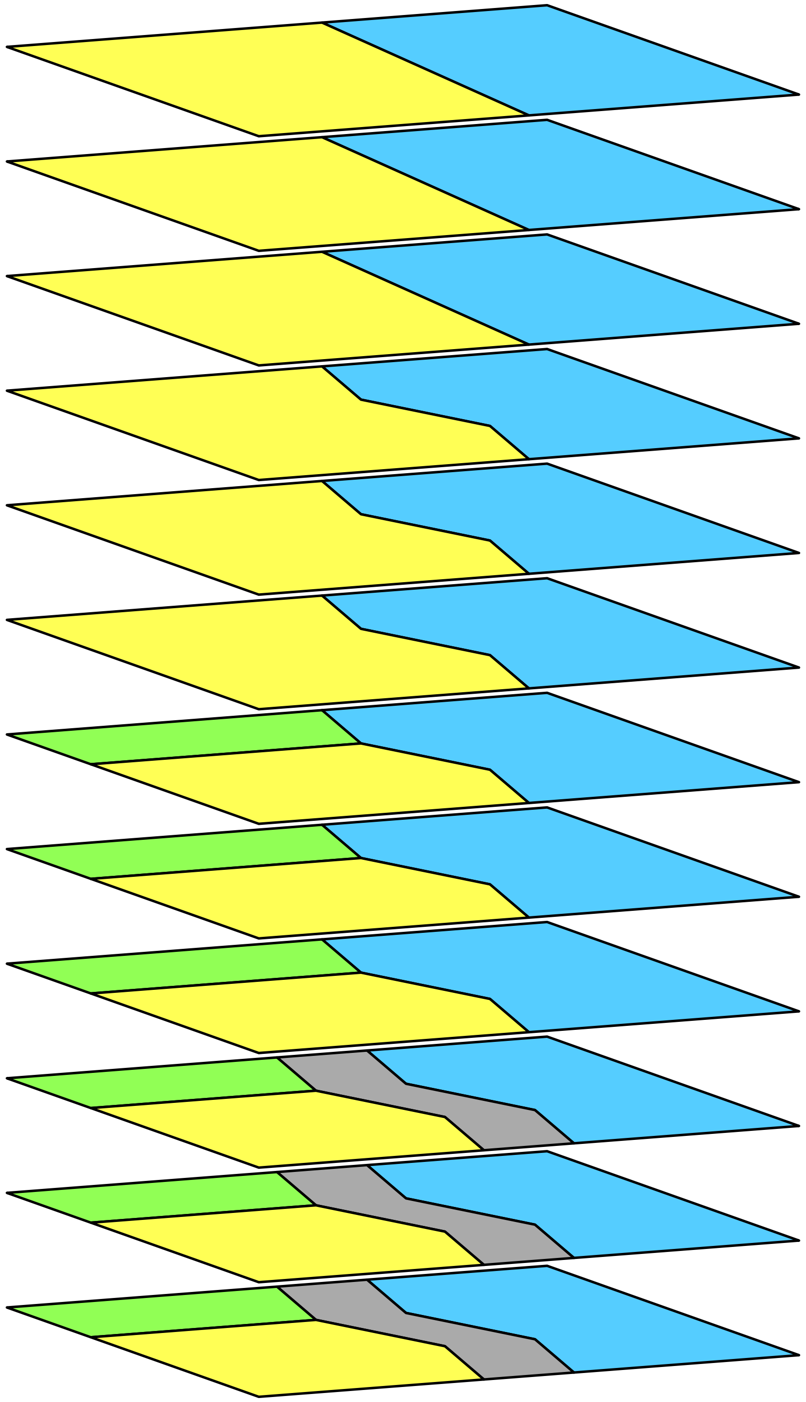
2D + scale

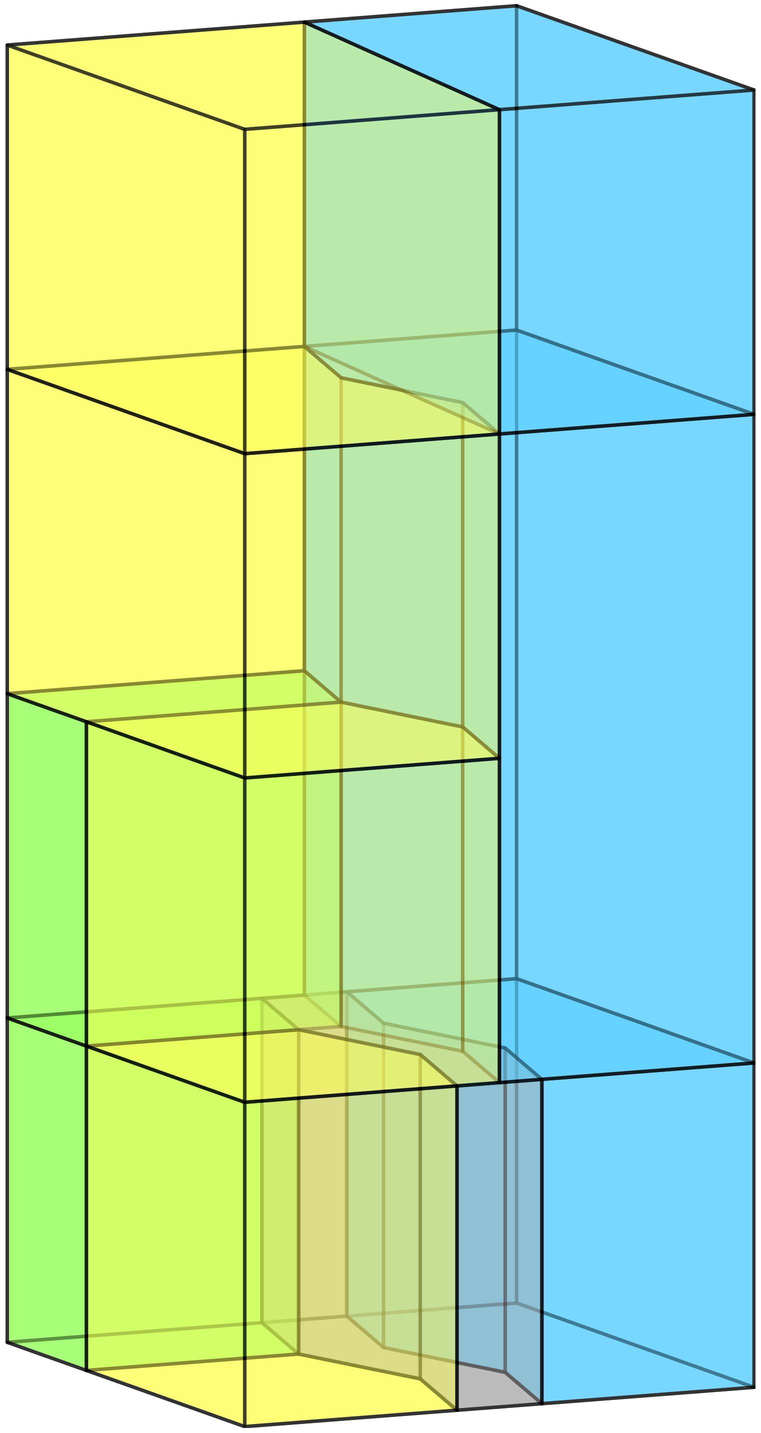
Land cover datasets



2D + scale

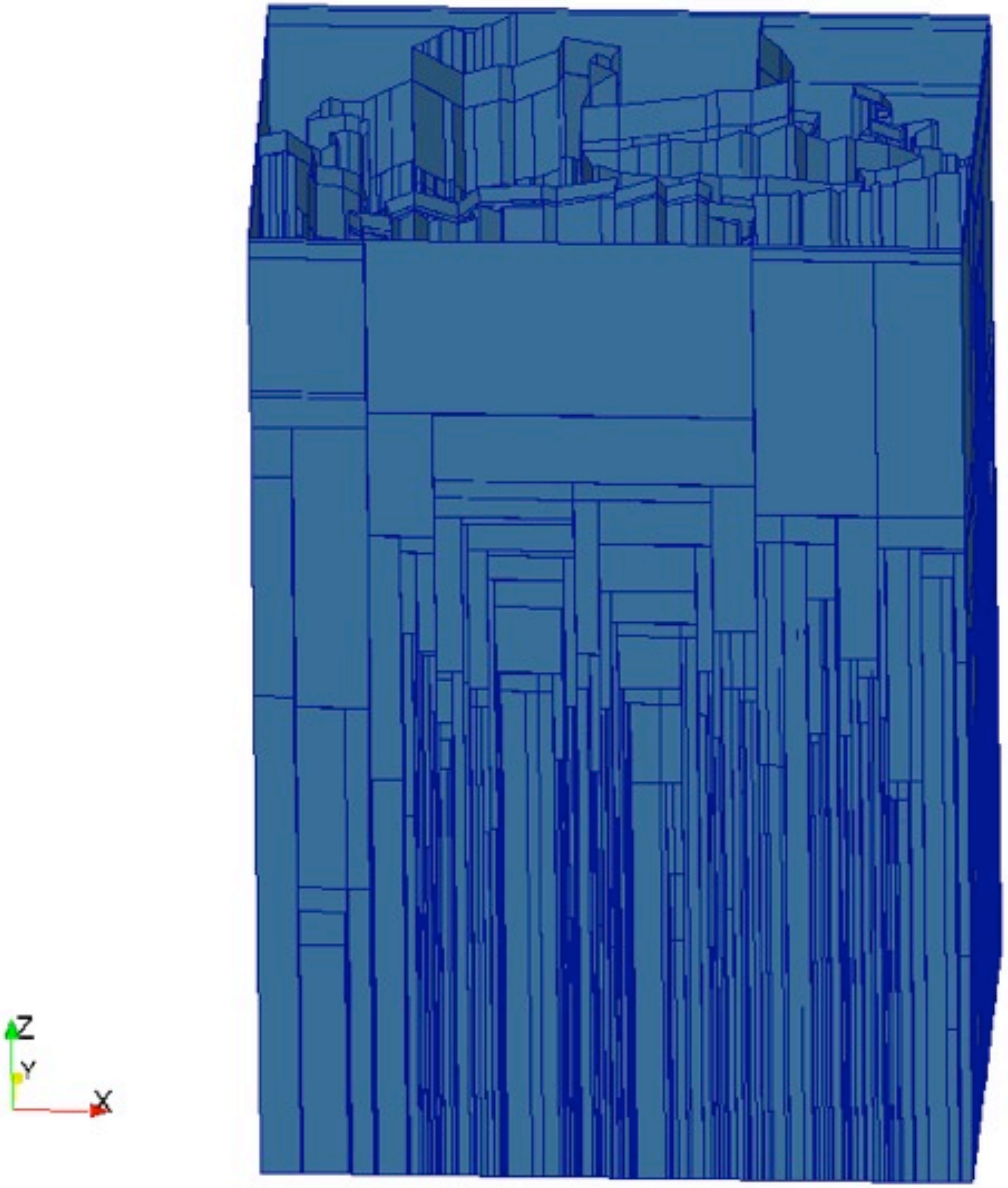
Graph of generalisation process





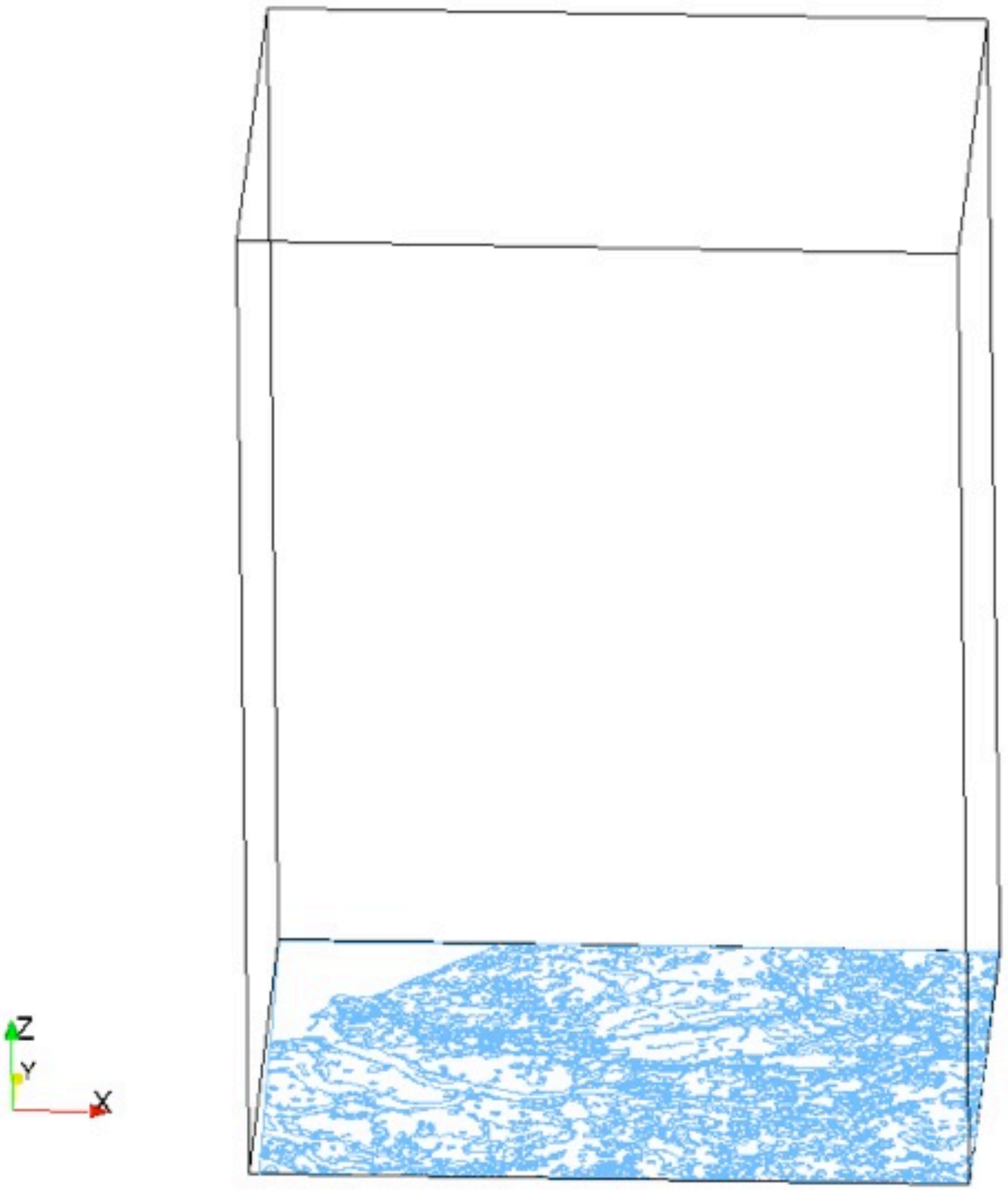
2D + scale = 3D model

Variable scale



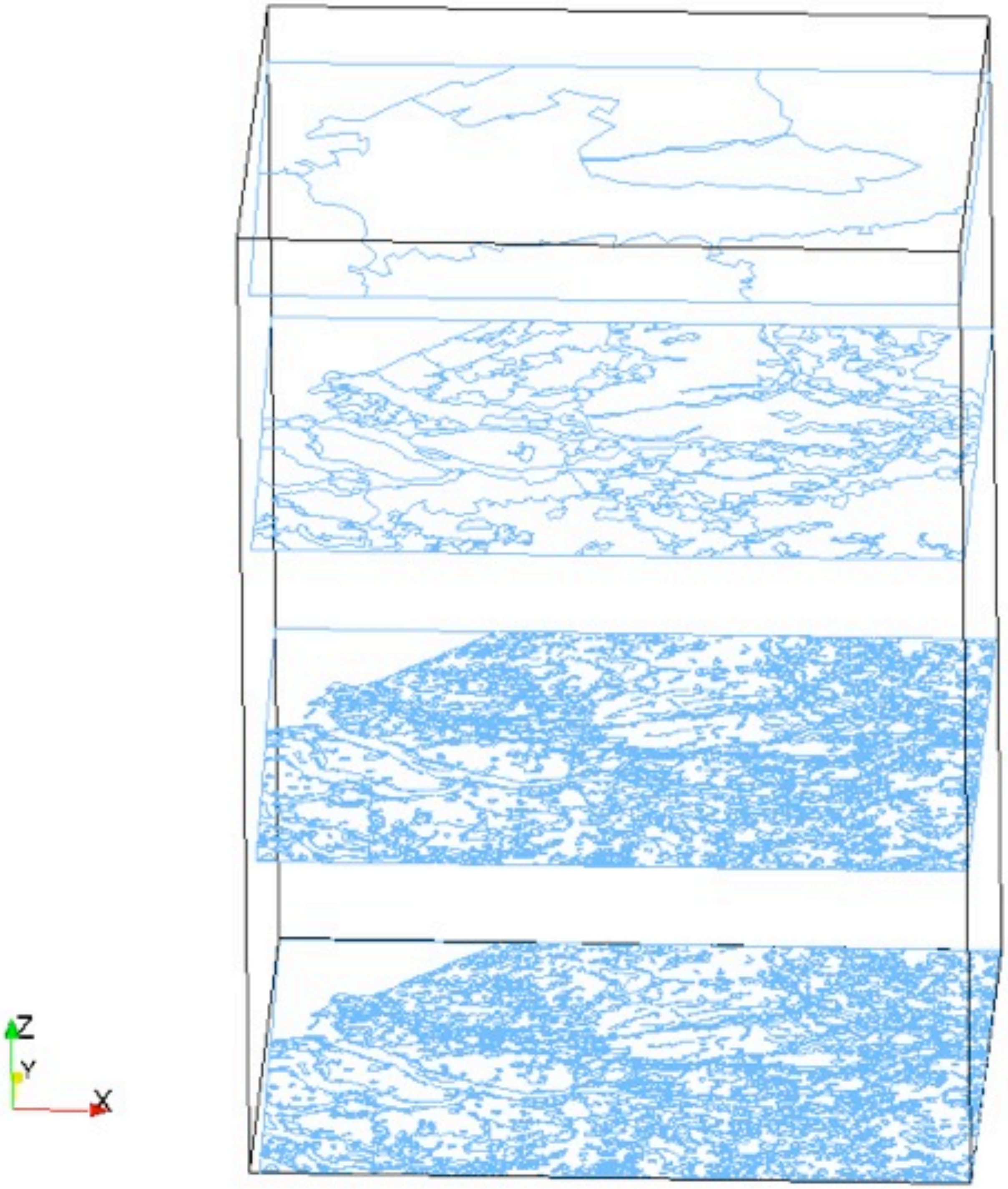
Variable scale

No predefined scales



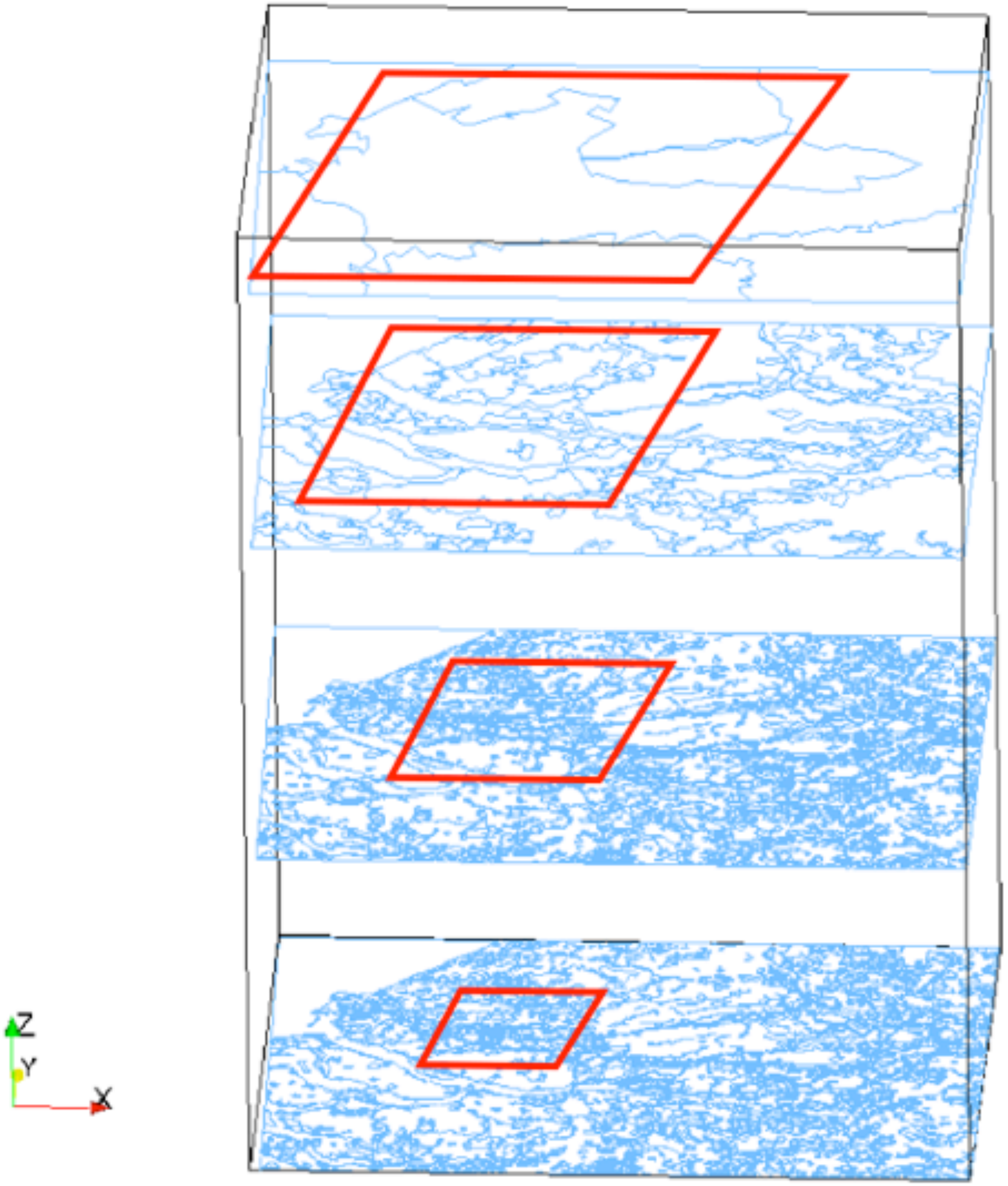
Variable scale

No predefined scales



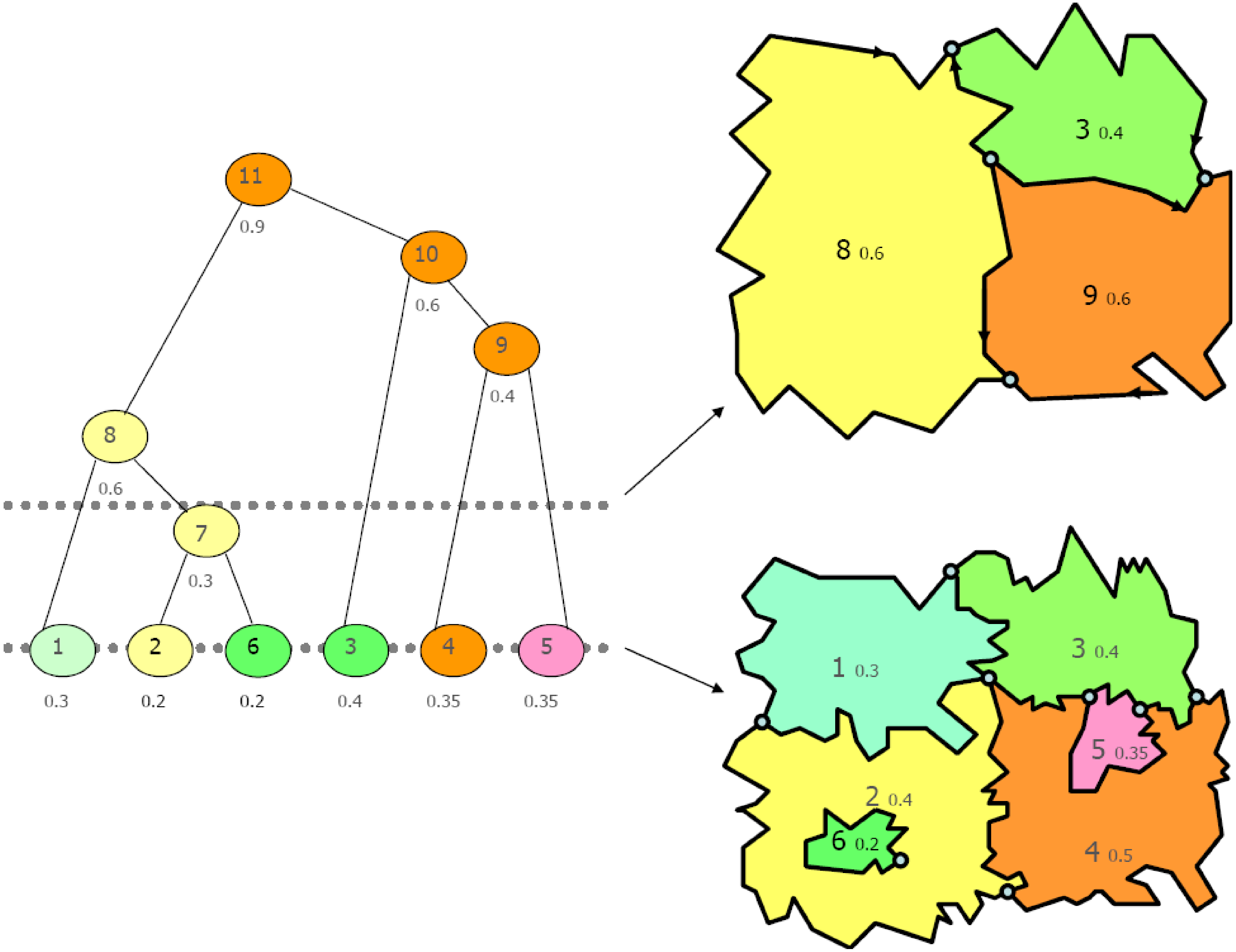
Variable scale

No predefined scales



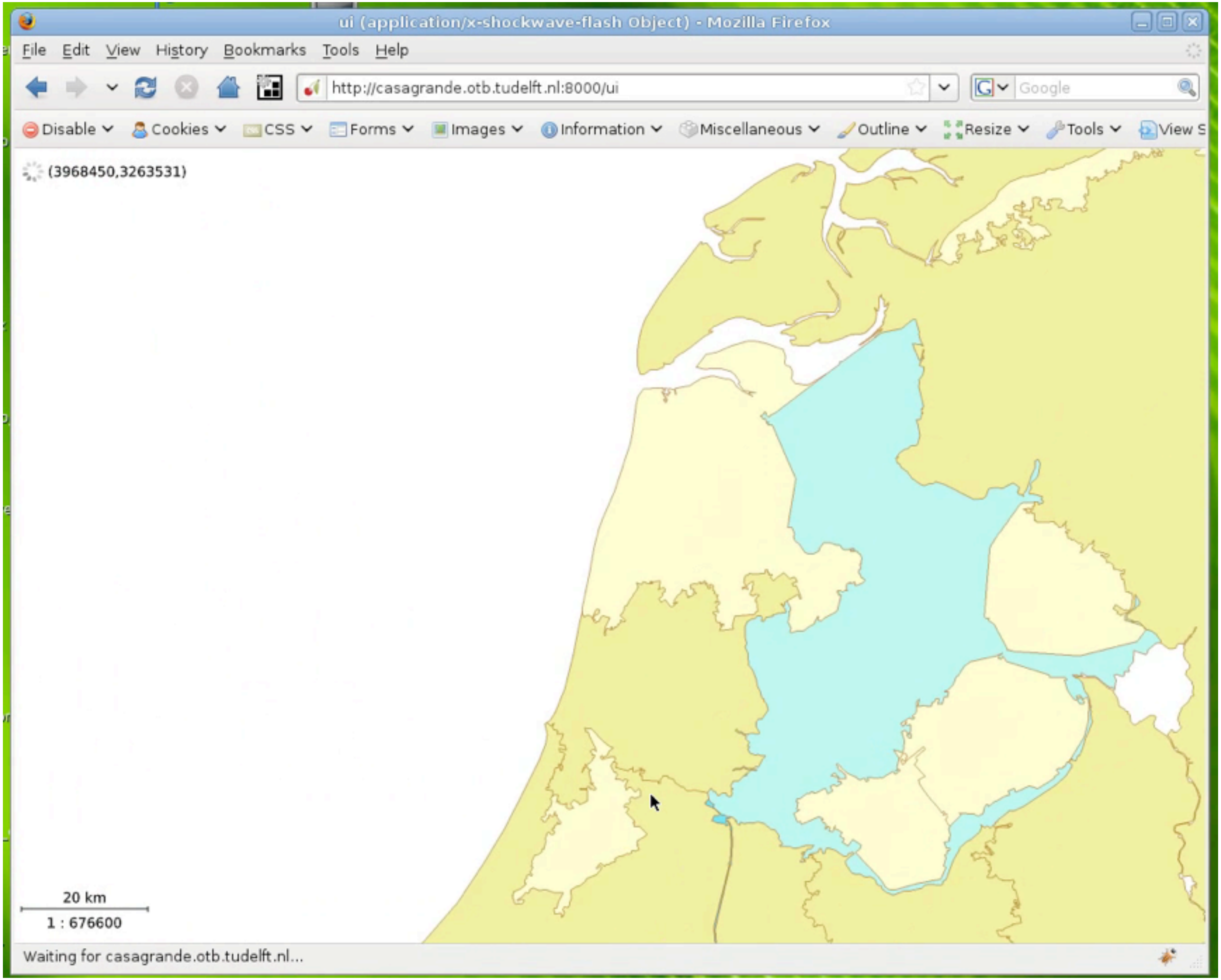
Variable scale

No predefined scales



tGAP

One possible data structure

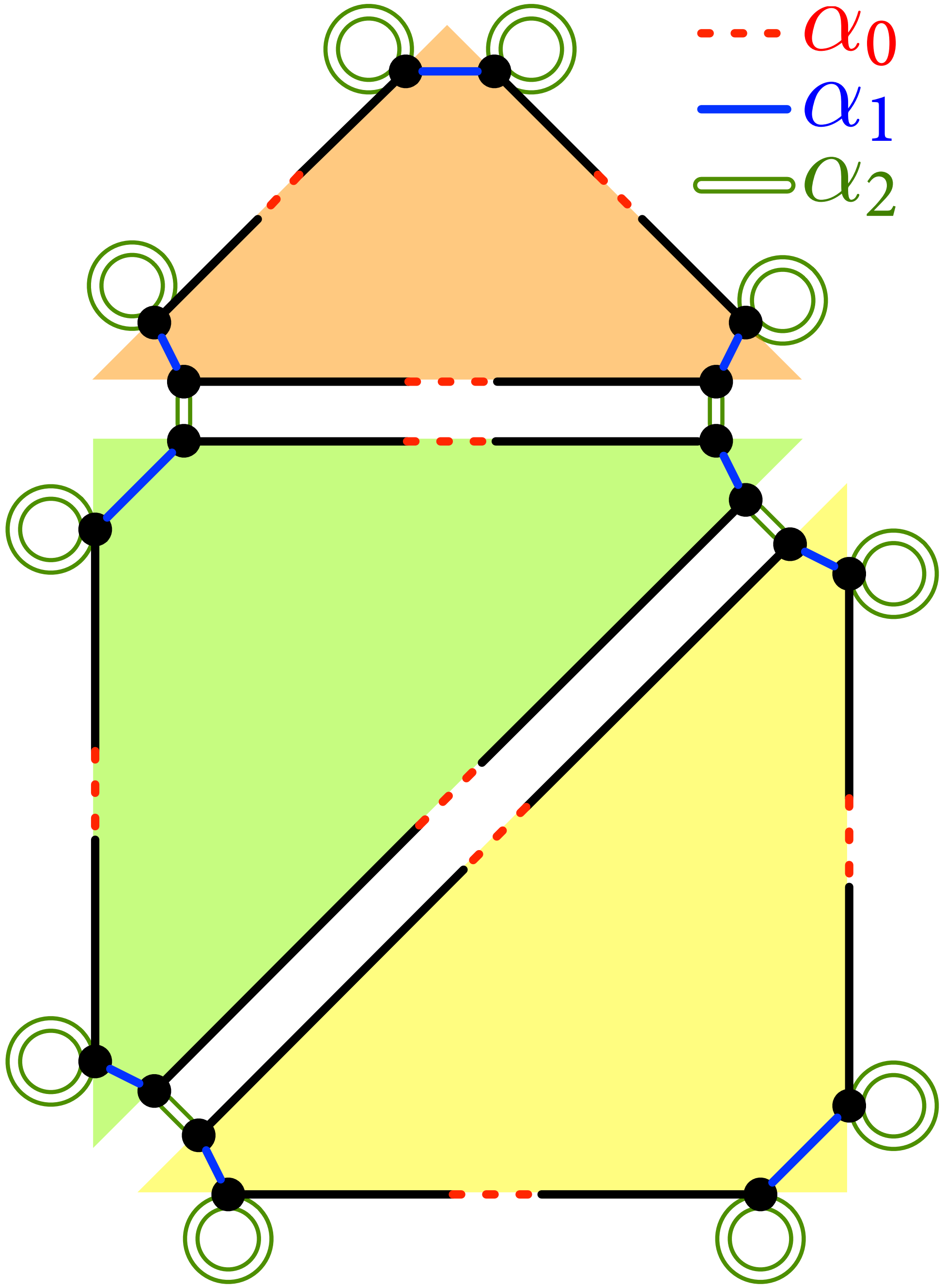


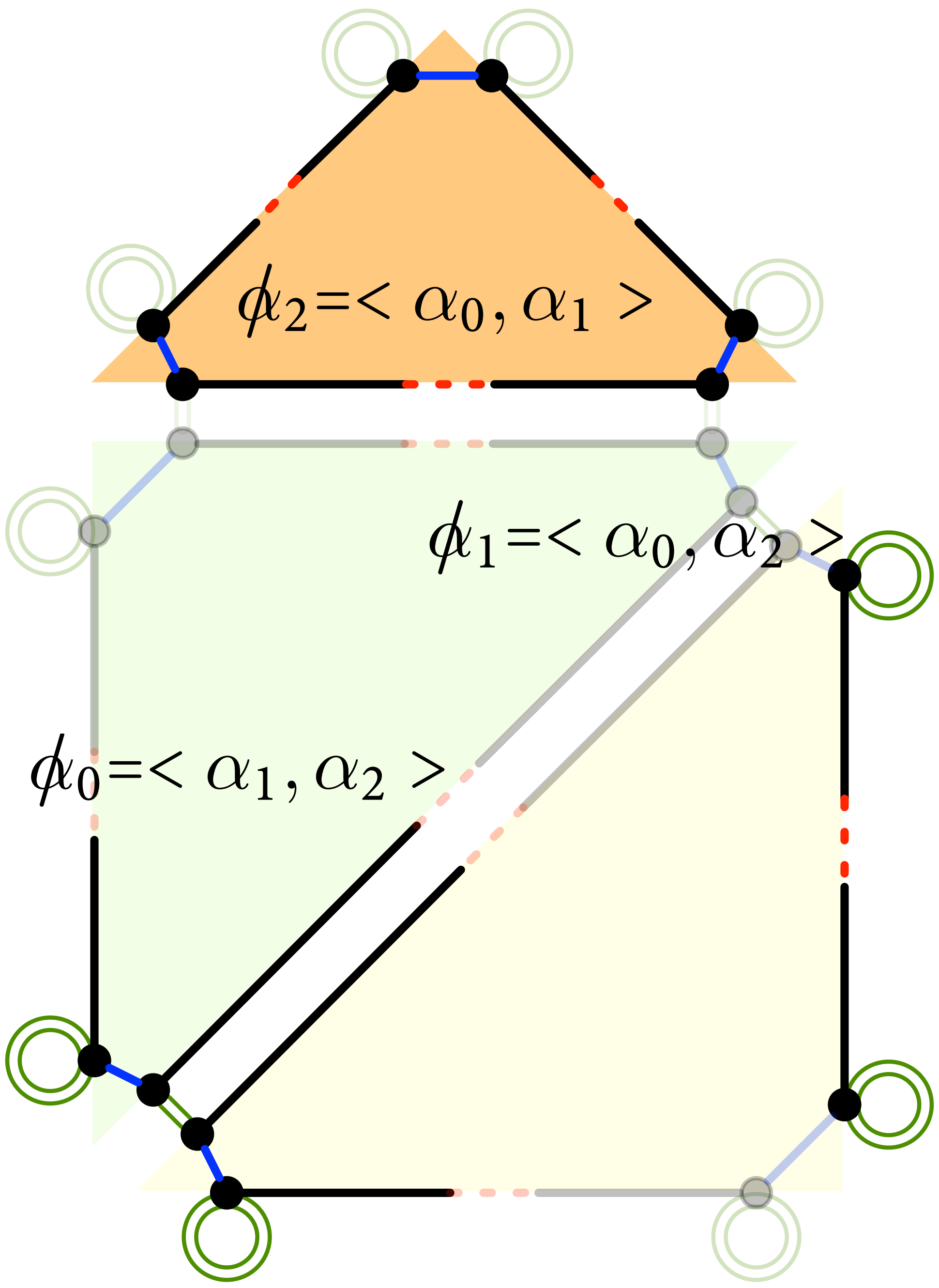
ui (application/x-shockwave-flash Object) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://casagrande.otb.tudelft.nl:8000/ui
Google
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View S
20 km
1 : 676600

(3968450,3263531)

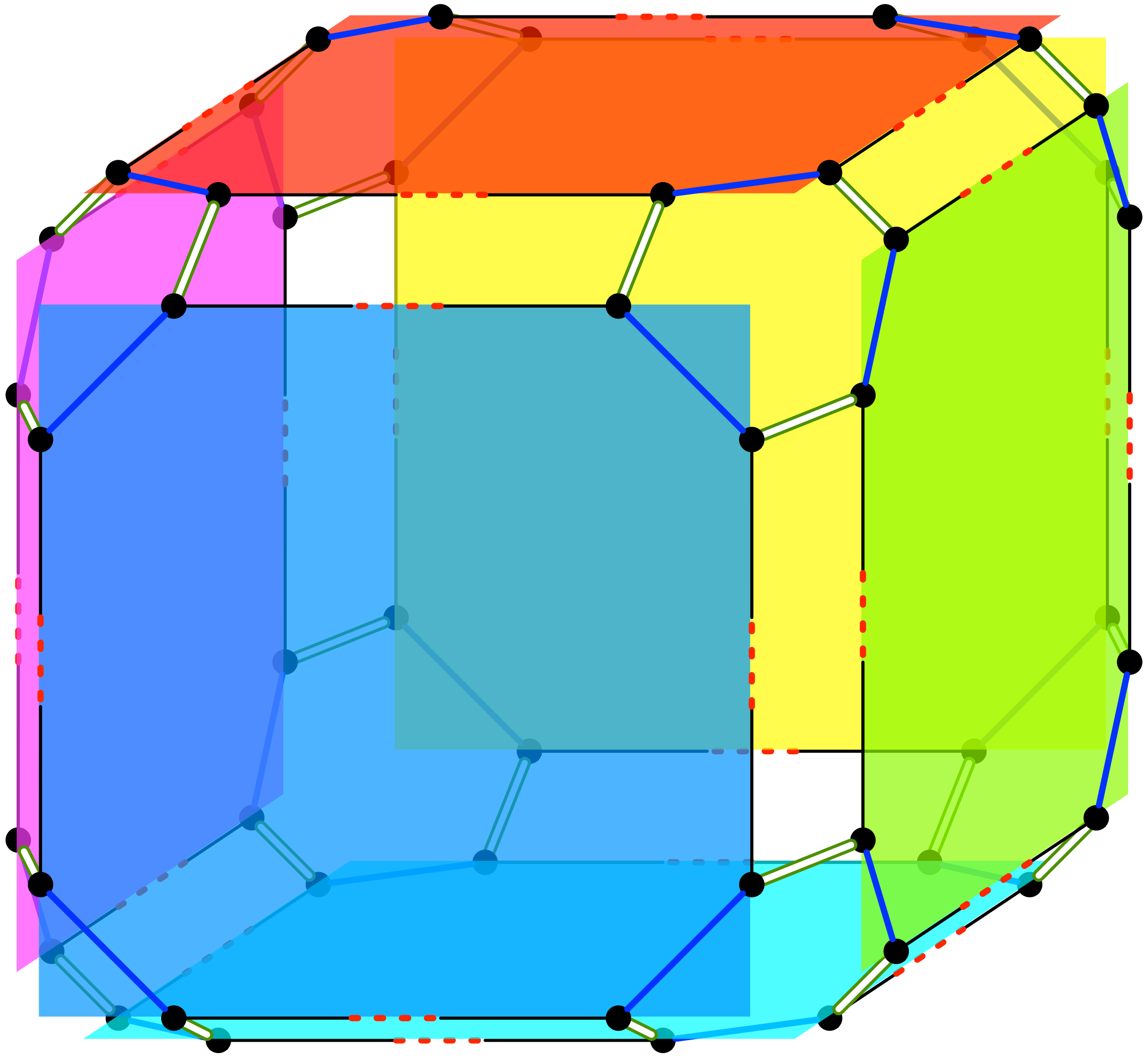
Waiting for casagrande.otb.tudelft.nl...

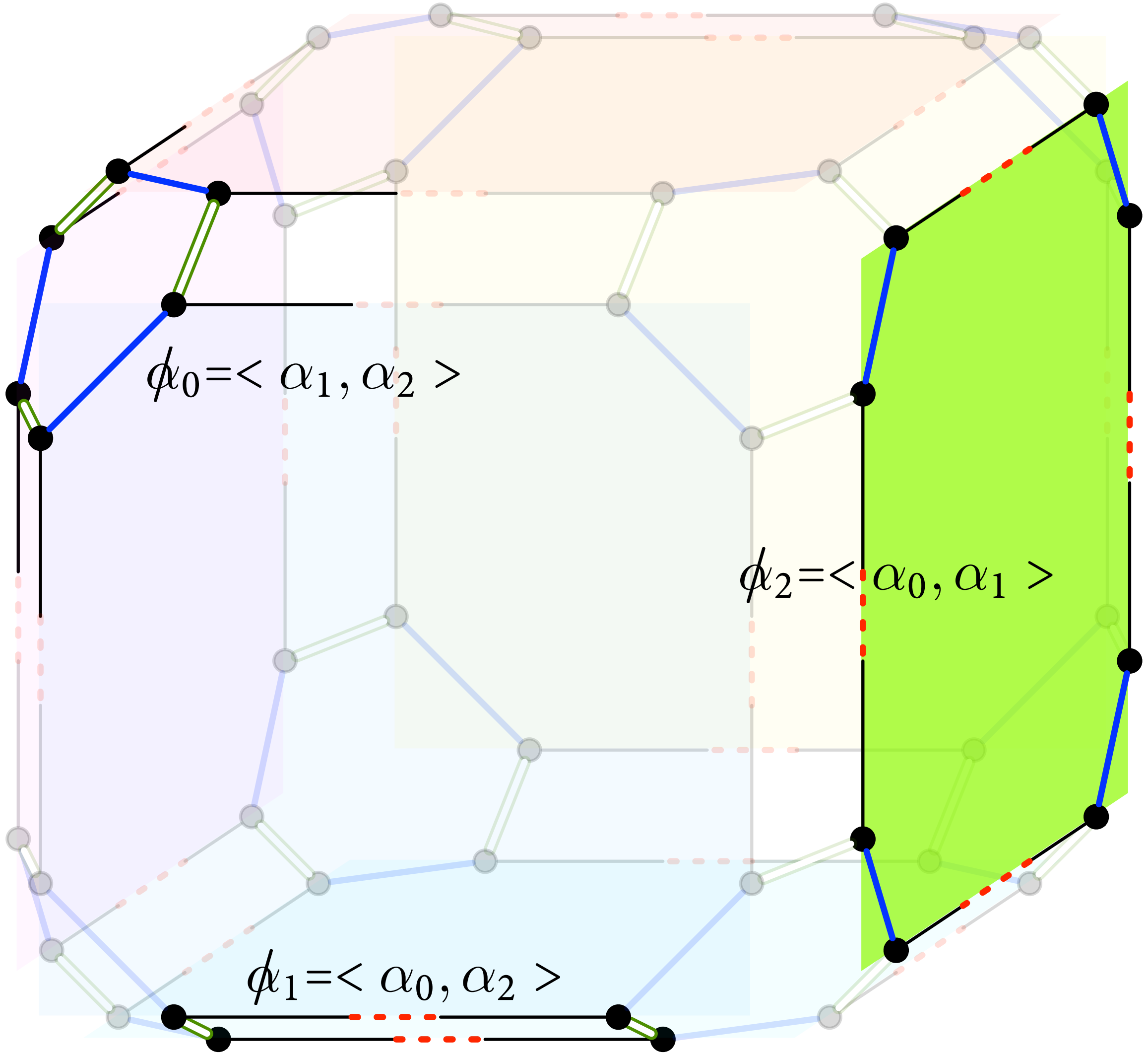
How to build and represent a 4D cube?


α_0
α_1
α_2


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

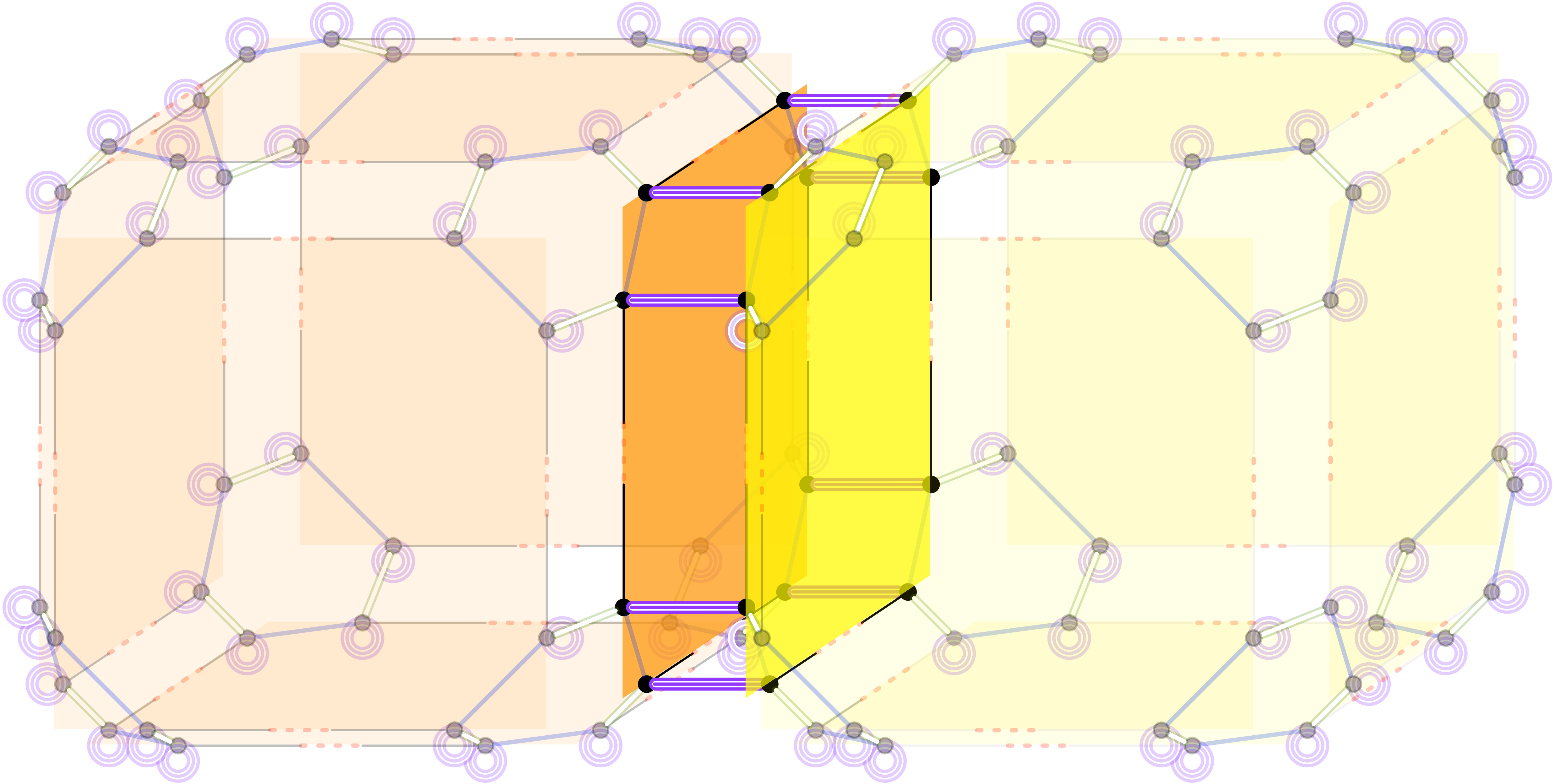
G-maps | 2D




ϕ_0=< α_1, α_2 >
ϕ_2=< α_0, α_1 >
ϕ_1=< α_0, α_2 >

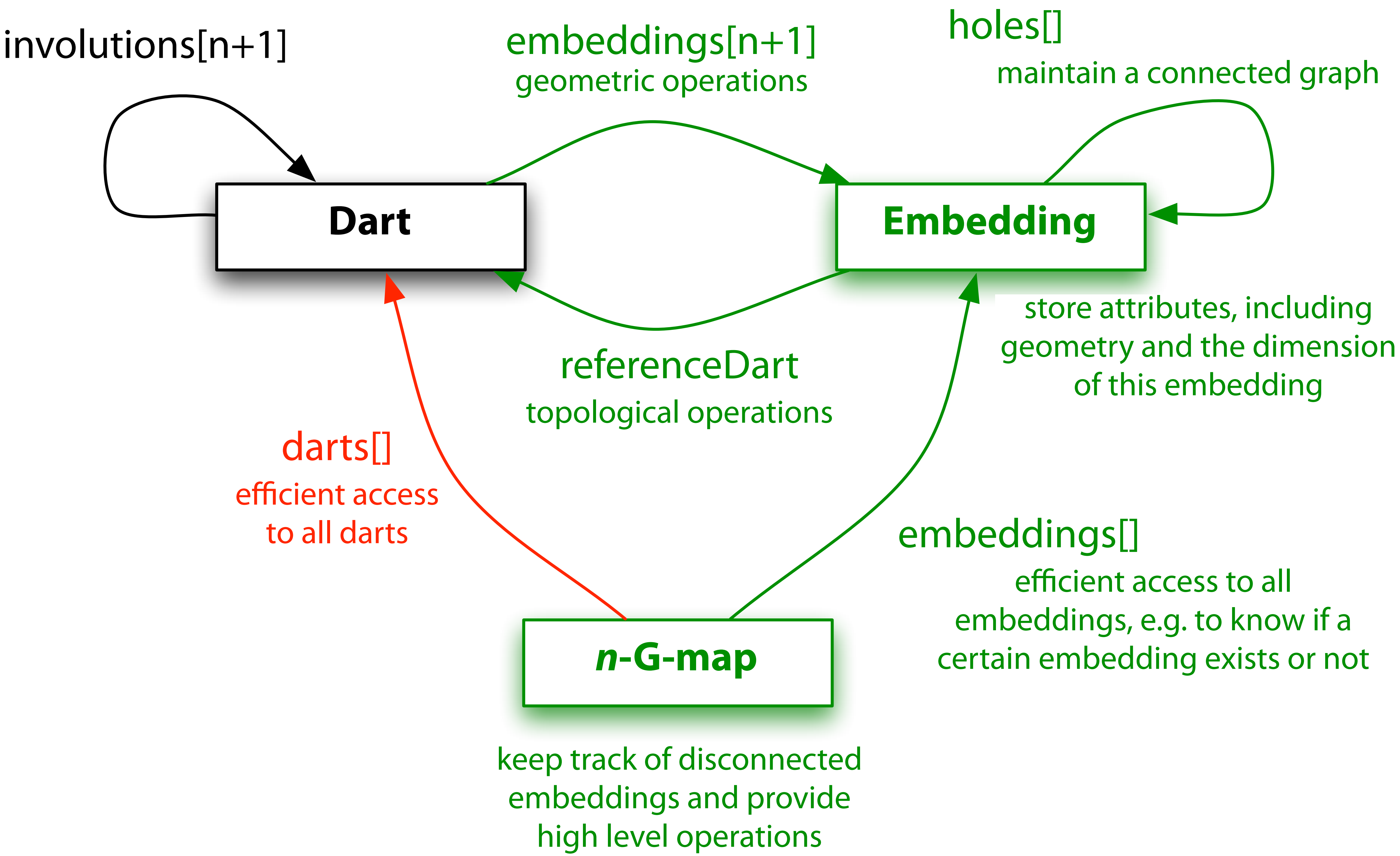
G-maps

3D



G-maps

3D



involutions[n+1]

embeddings[n+1]

geometric operations

holes[]

maintain a connected graph

Dart

Embedding

store attributes, including geometry and the dimension of this embedding

referenceDart

topological operations

darts[]

efficient access to all darts

embeddings[]

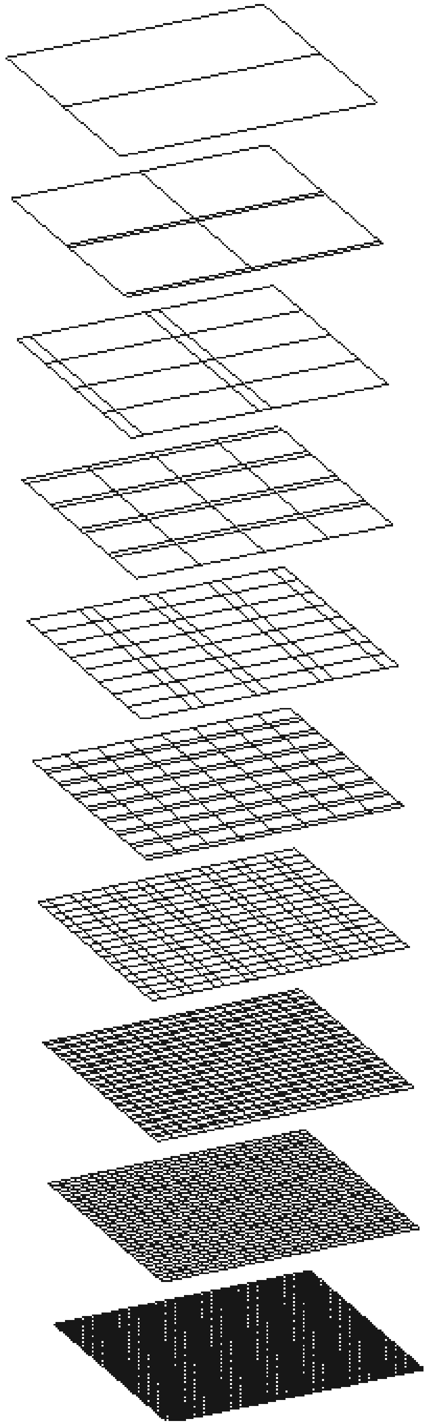
efficient access to all embeddings, e.g. to know if a certain embedding exists or not

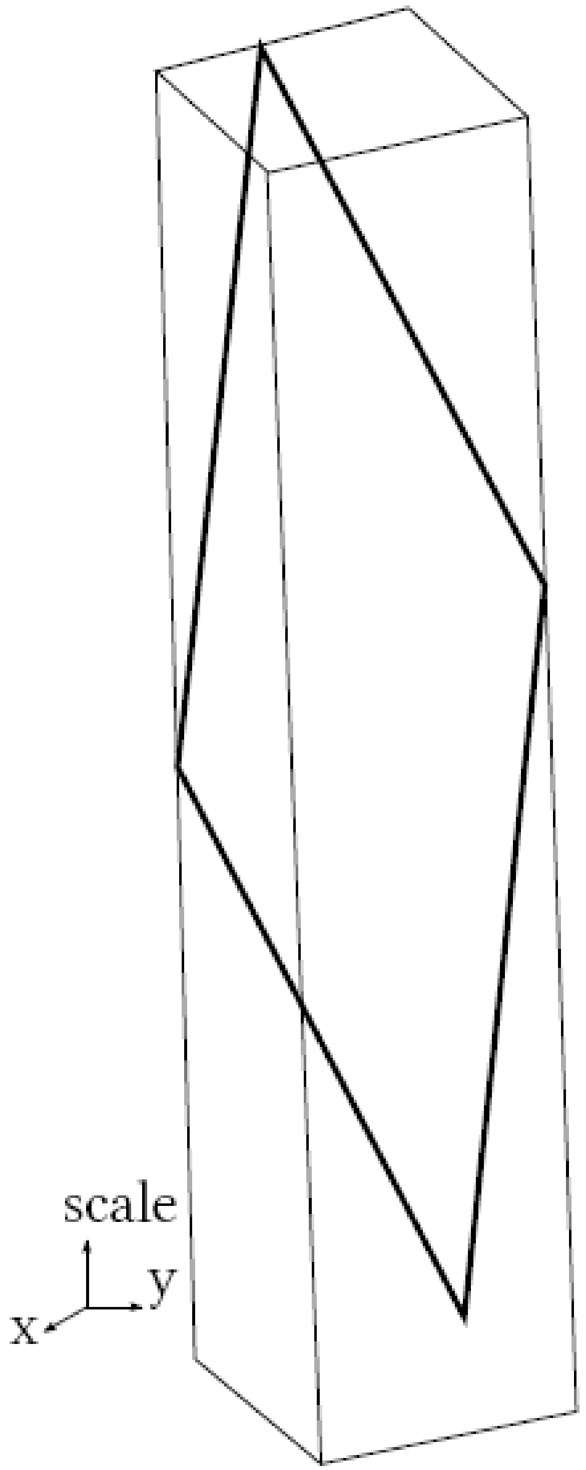
***n*-G-map**

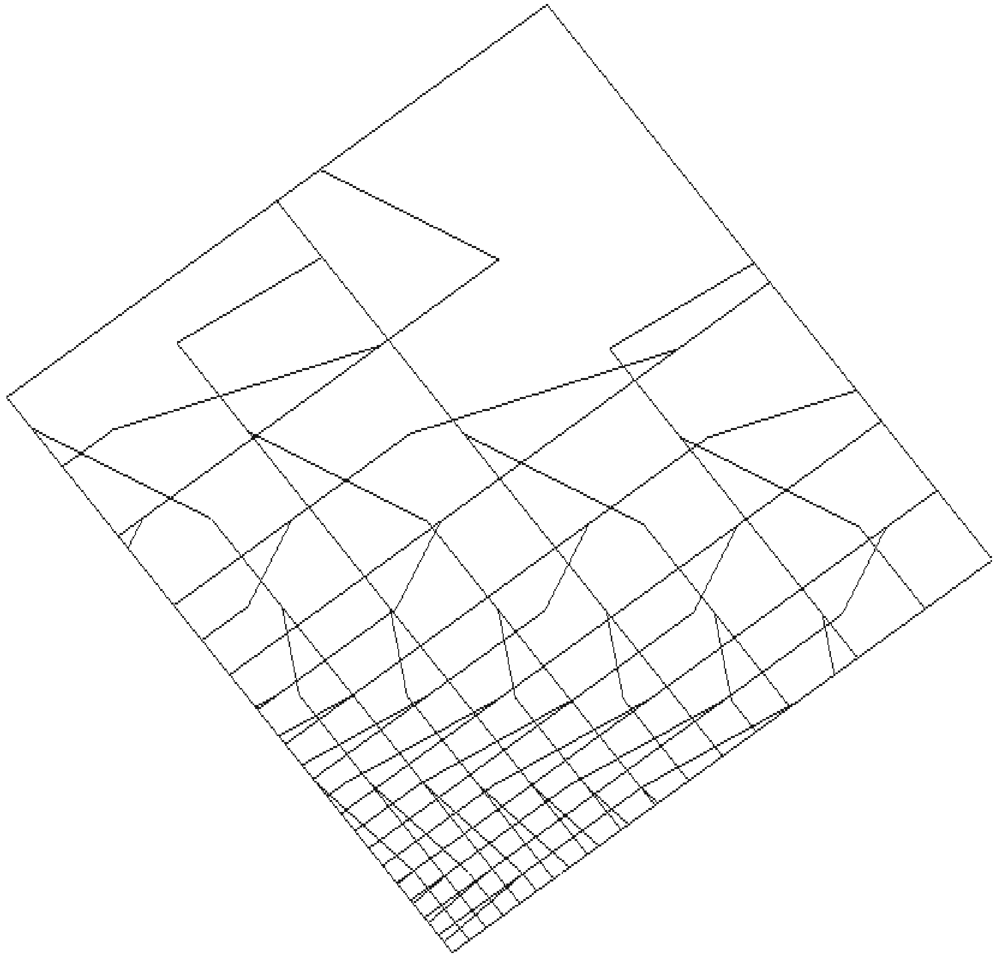
keep track of disconnected embeddings and provide high level operations

3D city models + LODs in one hypercube

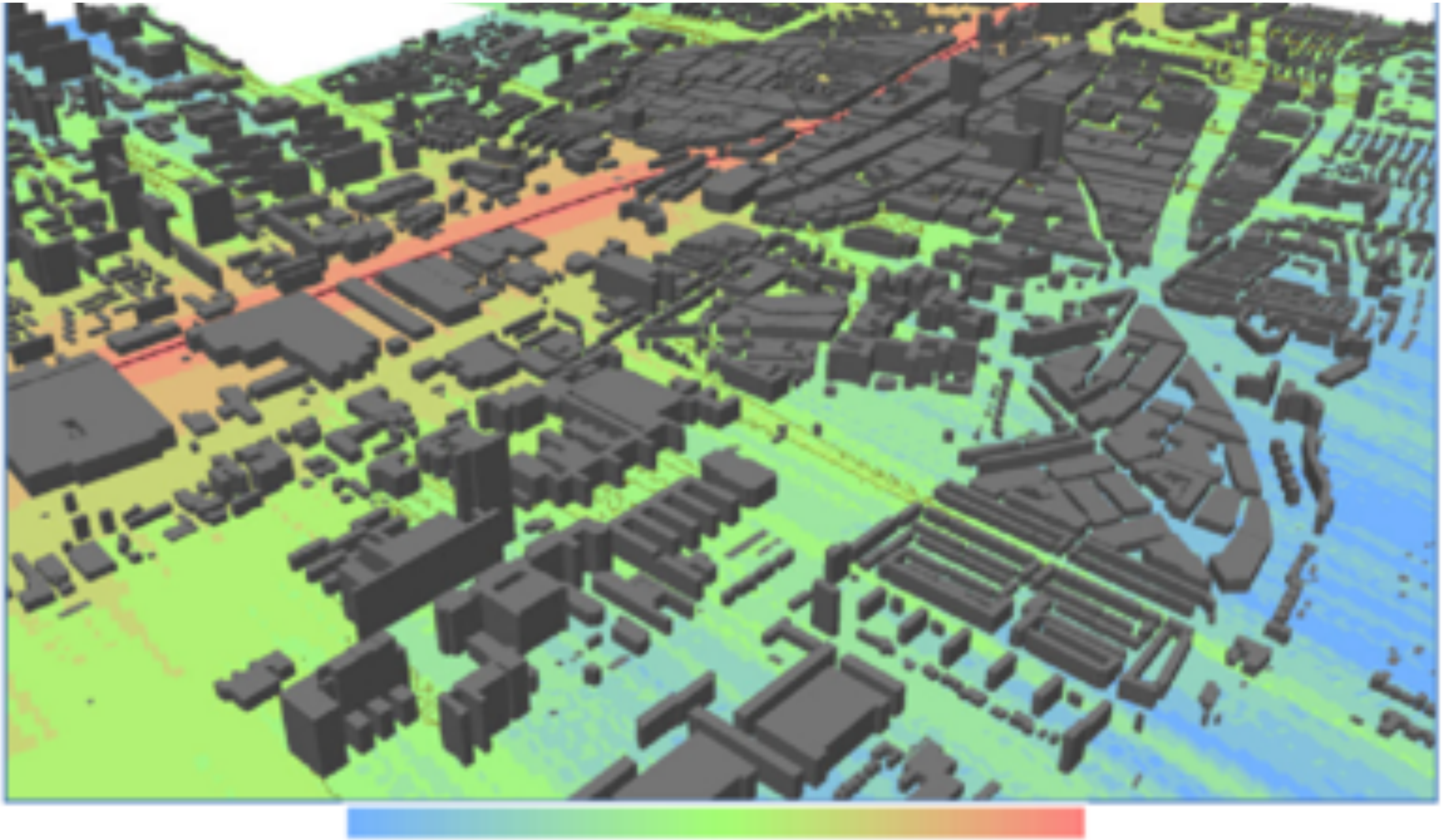
- Start with LOD3 and perform automatic generalisation to populate the cube
- Storing with G-maps data structure
- We can ensure consistency across the different LODs
- Slicing to obtain one 3D model





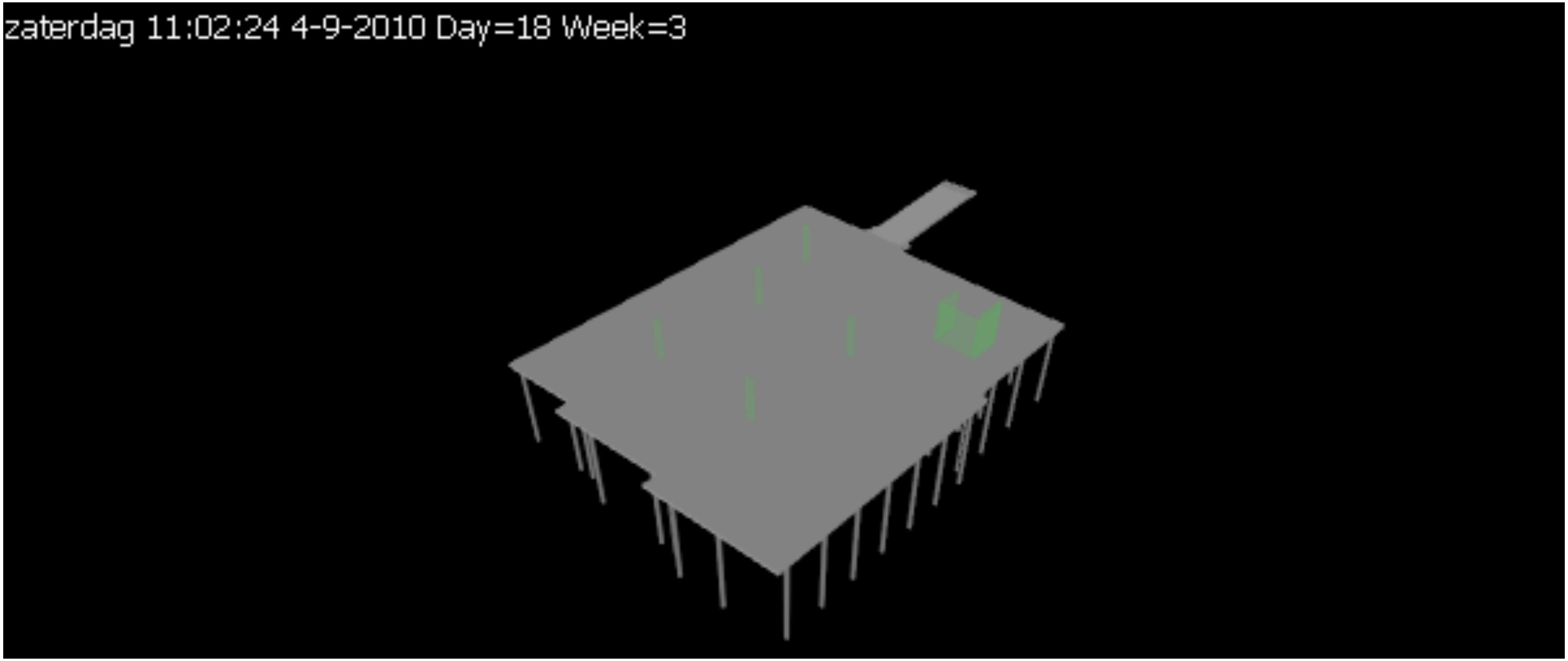


Non-horizontal slicing = mixed scale



Modelling of noise along the railway

More details closer to railway



4D space-time example

IFC file of building's construction

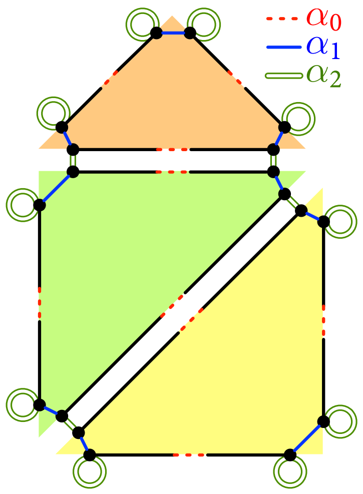
Thank you.

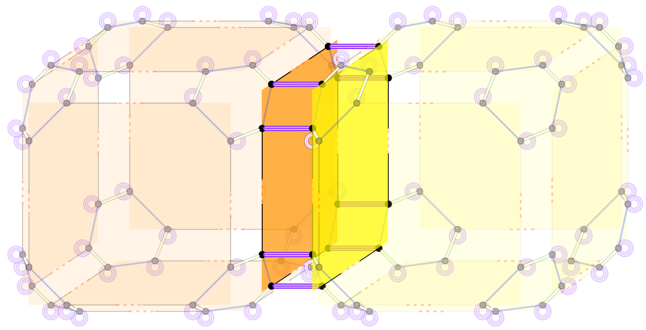
```
struct Dart {
    Dart *involutions[n+1];
    Embeddings *embeddings[n+1];
};

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

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

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

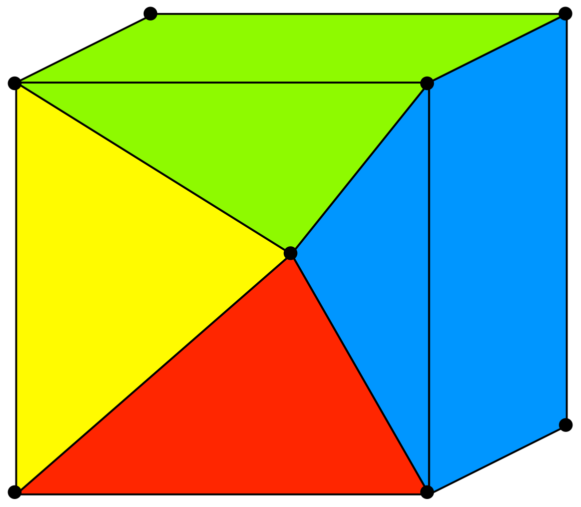
What is needed then?

Memory consumption

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







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