Cubical Binary Codes

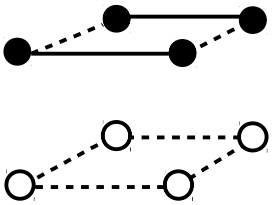
by Sven Nilsen, 2019

In this paper I introduce a way to address parts of n-dimensional cubes and their types with 2n bits.

A cubical binary code addresses parts of n-dimensional cubes and their types with only 2n bits.

- The first part refers to the type of the part. This takes `n` bits.
- The second part selects the part. This takes `n – count_ones(type)` bits.
- The third part refers to the sub-type. This takes `n – count_zeros(type)` bits.

For example, a 3D cube can refer to `$2^6$ = 64` different parts.



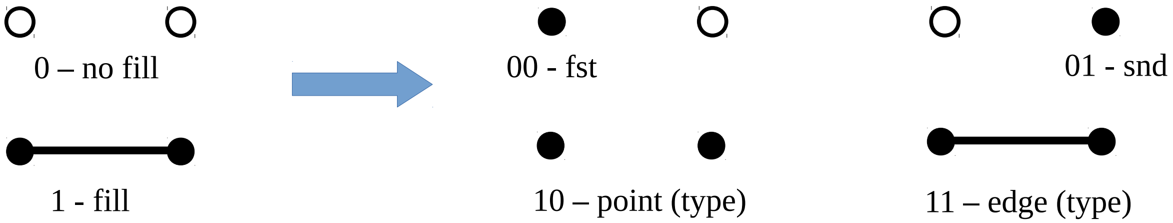
The x-edges of the y-most surface along zx plane: ***101101***
101 – surfaces along xz   1 – y-most surface   01 – x-edges

A 3D cube has the following binary code:

000 – points of cube,	3 bits selecting the point,	0 bits for sub-type
001 – edges along x	2 bits selecting the edge,	1 bit for sub-type
010 – edges along y	…	…
100 – edges along z	…	…
110 – surfaces along yz	1 bit selecting the surface	2 bits for sub-type
101 – surfaces along zx	…	…
011 – surfaces along xy	…	...
111 – type	0 bits	3 bits for sub-type

The volume of a 3D cube is encoded `111 111`.
The first 3 bits tells that this is a type, the 3 next bits tell that this is the volume.

The basic building block of this code is to build up from fill/no fill:


0 – no fill
00 - fst
01 - snd
1 - fill
10 – point (type)
11 – edge (type)