THE STRUCTURE OF A CUBE

The *key idea* is that the structure (geometry) of an object is an intrinsic property. Structure should make no reference to external relations.

Note that translation, rotation, scale, and orientation are Relations between an object and an external coordinate system, and are thus not part of a cube's geometry.

Fortunately, there are established conceptual tools (Cartesian geometry, unit vectors) for describing "cube space".

EMBED THE CUBE IN A SPACE

Assume unit vectors i, j, and k. Associate each with an orthogonal side of the Cube.

Given rules for ijk: i*j = i*k = j*k = 0

Assume a local origin (0i 0j 0k).

```
i = (1i 0j 0k)
j = (0i 1j 0k)
k = (0i 0j 1k)
```

DIFFERENTIATE PARTS

Cubes have 27 parts: 8 vertices, 12 edges, 6 faces, 1 volume.

Notation: (ai bj ck) for all parts.

Let {a, b, c} take on three possible states: {0, _, 1},

where _ is any value 0 =< _ =< 1

Let d = {0, 1} (Knonecker delta, either 0 or 1)

```
Vertices:  {di dj dk}
Edges:     {di dj _k} or {di _j dk} or {_i dj dk}
Faces:     {di _j _k} or {_i dj _k} or {_i _j dk}
Solid:     {_i _j _k}
```

More notation:

Let i, j, and k be symmetrically equivalent, and thus unlabeled.

Vertices:	{d d d}	(all three states are Kronecker)
Edges:	{d d _}	(one state is not Kronecker)
Faces:	{d _ _}	(only one state is Kronecker)
Solid:	{_ _ _}	(no state is Kronecker)

Let u stand for any of i, j, or k.

PROPERTIES

Parallel(e1_ e2_) =	e1{d d _} = e2{d d _}	_ in same location
Parallel(f1_ f2_) =	f1{d _ _} = f2{d _ _}	_ _ in same location

Perpendicular(e1_ e2_) =	not(Parallel(e1 e2))
Perpendicular(f1_ f2_) =	not(Parallel(f1 f2))

On(v_ e_) =	v{du} = e{du}	values of d equal
On(v_ f_) =	v{du} = f{du}	value of d equal
On(e_ f_) =	e{du} = f{du}	value of d equal

Meets(e1_ e2_) =	e1{du} = e2{du}	some d equal
Meets(f1_ f2_) =	not(Parallel(f1 f2))	

Distance(v1_ v2_) =	number of different {du}
Distance(e1_ e2_) =	number of different {du}

PICTORIALLY

```
                      011  _11  111
                      0_1  __1  1_1
                      001  _01  101

                 01_  _1_  11_
                 0__  ___  1__
                 00_  _0_  10_

            010  _10  110
            0_0  __0  1_0
            000  _00  100
```

```
                                                back = __1
                                    top = _1_


                        011 ----------_11------------  111
                         / |                        / |
                        /  |                       /  |
                       /   |                      /   |
                     01_   |                    11_   |
                     /     |                     /    |
                    /     0_1                  /     1_1
                   /       |                  /       |     rside = 1__
               010 -----------_10-----------  110     |
                  |        |                 |        |
                  |        |                 |        |
 lside = 0__      |        |                 |        |
                  |     001 ----------_01----|-------  101
                  |       /                  |       /
                 0_0     /                  1_0     /
                  |     /                    |     /
                  |   00_                    |   10_
                  |   /                      |   /
                  |  /                       |  /
                  | /                        | /
               000 -----------_00-----------  100


                             bottom = _0_
               front = __0

                                                            solid = ___
```

MULTIPLICATION TABLES

To determine vertex of intersection of two edges (or edge of intersection of two faces, or more generally, lower dimensional element defined by two other elements), down-multiply representation:

*	0	_	1
0	0	0	_
_	0	_	1
1	_	1	1

To determine edge formed by two vertices (or general up element), up-multiply representations:

%	0	_	1
0	0	_	_
_	_	_	_
1	_	_	1

Note than non-intersecting vertices identify faces (or solids)

NOTES ON REPRESENTATION

By multiplying i, j, or k by a scalar, the cube generalizes to an arbitrary block.

ijk provides lots of established mathematical support.

{0 _ 1} provides unification of different parts of a cube and visual imagery.

Binary Kronecker delta provides easy implementation, but could be renamed (0 = low, 1 = high, _ = any) for understanding.

Properties are trivial calculations.

Generality of notation is difficult to express algebraically. In general, the more abstract, the more powerful and the harder to express.