Table filling transforms

hard-code once:

2^4 table entries: 12 edges, 2 colors

* how many permutations are possible? (so both have same case)

\[ \frac{8 \cdot 3 \cdot 2}{8} = 48 \]

* need a simple, small set of transformation to generate all of them

  \[ \text{rotate } x \text{ axis (or} y \text{ or} z) \]

  \[ \text{rotate } \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \text{ (about diagonal)} \]

  \[ \text{flip } x \rightarrow -x \text{ (or} y \text{ or} z) \]

  \[ \text{color inversion} \]

  \[ \text{need rot + flip + color} \]

  \[ \text{eq: rot-x rot-y flip-x color-swap} \]

  \[ \text{just needs to generate everything, does not need to be efficient} \]

  \[ \text{(but rotations preferred)} \]

algorithm

\[
\text{fill (case c, triangulations)} \\
\text{if not done, return false, done(false)} \\
\text{fill(c, t), done = true} \\
\text{fill(c, t1), (c2, t2) = rot x (c, t1)} \\
\text{fill(c2, t2), (c3, t3) = rot y (c2, t2)} \\
\text{fill(c3, t3), (c4, t4) = flip x (c3, t3)} \\
\text{fill(c4, t4)} \\
\text{if not analogous (c4, t4) = flip color (c, t); fill(c, t4)} 
\]
* Marching cubes ambiguity

ambiguous face
both cubes must agree on us

* green connected

* red connected

* many ways to resolve. e.g., always connect the red vertices

* when filling the table, some transformations do not preserve the resolution

ok
rotation
flip

breaks resolution
color flip

* must create both triangulations for these:

but not these:

* check for ambiguous face before doing color flip
Torquation and orientation

* CCW viewed from outside (o = inside)

* Transformations may reverse this!

→ preserve rotate
flip orientation
flip color flip
consistent
Other notes

* Choose edge numbering wisely - it helps!
* For each transform, make a lookup table mapping:
  1. old vertex to new vertex
  2. old edge to new edge
* Use table 1 to compute the new case
* Use table 2 to compute the new triangulation
* No tables needed for color flip

* Compact representation:
  - 5 triangles (max)
  - 3 vertices per triangle (each an edge of cube)
  - 12 edges → 4 bits
  - 4 · 3 · 5 = 60 bits
  - 64-bit log per case encodes everything!
  - 64-bit = 8 byte · 256 cases = 2KB table (small!)