# How to Make Marching Squares

Aresh Pourkavoos

January 7, 2022

Every "marching squares-"type algorithm is equivalent to a height-map problem one dimension up: create a height-map mesh in $n+1$ dimensions, and *intersect it with a hyperplane at the threshold height.* For marching cubes, this would be a mesh in 4D made out of tetrahedra. A triangular grid in 2D makes this trivial, and for a square grid, each square can be split into 4 triangles by adding the average value to the center.

Transvoxel (Eric Lengyel) can be designed much more easily using this principle. As the threshold changes, the mesh will vary continuously, since the higher-dimensional mesh has no gaps - no need to consider how one case transitions into another.

Edge case: an entire facet of the mesh (triangle/tetrahedron/etc) lies on the slicing hyperplane - cut in the middle somehow, or perturb the vertices to be slightly above/below threshold

Square needs to be dissected into triangles, cube into tetrahedra: square can be 2 (minimum) or 4 (symmetry-preserving), cube can be 4 (minimum) or 24 (symmetry-preserving)

How to ensure that the orientation of the simplices is consistent? Does the higher-dim mesh's consistency ensure it?