About Hilbert's Cube Filling Curve*

See also: The more famous Hilbert SquareFillCurve.

Hilbert's cube filling curve is a continuous curve whose image fills a cube. It is a straight forward generalization of the continuous square filling curve. It is shown in anaglyph stereo via a sequence of polygonal approximations. Each approximation is a polygon that joins two neighboring vertices of the cube.
The iteration step goes as follows:
The cube with the given (initial or a later) approximation is scaled with the factor 1/2. Eight of these smaller copies are put together so that they again make up the original cube, and this is done in such a way that the endpoint of the curve in the first cube and the initial point of the curve in the second cube fit together, and so on with all eight cubes. The result of one iteration therefore is a curve that is four times as long as the previous curve and that runs more densely through the cube. In 3DXM, if one rotates the cube with the mouse then the cube and its first subdividing eight cubes are shown together with one iteration of the initial curve.

To acheive a better feeling for the iteration step, one can set the parameter cc to integer values between 0 and 5. This will select different initial curves. An even value of cc and

---

* This file is from the 3D-XplorMath project. Please see: http://3D-XplorMath.org/

the following odd value give the same initial curve, but for even cc the Hilbert iteration is done *without* the endpoints, while for odd cc the endpoints are *included* in the iteration. (Using the Action Menu, one can switch between Hilbert's default (cc=0) and a case that emphasizes the iteration of the endpoints, cc=5.)
We have the same situation as in the two-dimensional case: The endpoints and their iterates are points that already lie on the limit curve because they are not changed under further iterations. One can say that the endpoints and their iterates are related to the limit curve in a very simple way. On the other hand, the approximating polygons develop double points at these iterates and the result is that the approximations look much more confusing if the endpoints and their iterates are included in the iteration. This is why we offer the choice between iterating with and without the endpoints.

H.K.