Algorithms of Scientific Computing

Space-Filling Curves in 3D

Tobias Neckel, Dirk Pflüger

Summer Term 2011
Classification of Space-filling Curves

**Definition:** (recursive space-filling curve)

A space-filling curve $f: I \to Q \subset \mathbb{R}^n$ is called **recursive**, if both $I$ and $Q$ can be divided in $m$ subintervals and subdomains, such that

- $f_*(I^{(\mu)}) = Q^{(\mu)}$ for all $\mu = 1, \ldots, m$, and
- all $Q^{(\mu)}$ are geometrically similar to $Q$.

**Definition:** (contiguous space-filling curve)

A recursive space-filling curve is called **contiguous**, if for any two neighbouring intervals $I^{(\nu)}$ and $I^{(\mu)}$ also the corresponding subdomains $Q^{(\nu)}$ and $Q^{(\mu)}$ are direct neighbours, i.e. share an $(n-1)$-dimensional hyperplane.
Contiguous, Recursive Space-filling Curves

Examples:
- all Hilbert curves (2D, 3D, …)
- all Peano curves

Properties: contiguous, recursive SFC are
- continuous (more exact: Hölder continuous with exponent $1/n$)
- neighbourship-preserving
- describable by a grammar
- describable in an arithmetic form
  (similar to that of the Hilbert curve)
3D Hilbert Curves

- Wanted: contiguous, recursive SFC, based on division-by-2
  - leads to 3 basic patterns:
    
    ![3D Hilbert Curves Diagram]

- in addition: symmetric forms, change of orientation
- always two different orientations of the components
  - numerous different Hilbert curves
3D Hilbert Curves – Iterations

1st iteration

2nd iteration
3D Hilbert Curve – Arithmetic Representation

t given in the octal system, \( t = 0_8.k_1k_2k_3k_4 \ldots \), then

\[
h(0_8.k_1k_2k_3k_4 \ldots) = H_{k_1} \circ H_{k_2} \circ H_{k_3} \circ H_{k_4} \circ \ldots \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}
\]

with operators

\[
H_0 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x + 0 \\ \frac{1}{2}z + 0 \\ \frac{1}{2}y + 0 \end{pmatrix} \quad \quad H_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}z + 0 \\ \frac{1}{2}y + \frac{1}{2} \\ \frac{1}{2}x + 0 \end{pmatrix}
\]

\[
H_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x + \frac{1}{2} \\ \frac{1}{2}y + \frac{1}{2} \\ \frac{1}{2}z + 0 \end{pmatrix} \quad \quad H_3 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}z + \frac{1}{2} \\ -\frac{1}{2}x + \frac{1}{2} \\ -\frac{1}{2}y + \frac{1}{2} \end{pmatrix}
\]
3D Hilbert Curve – Arithmetic Representation (cont.)

\[
H_4 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}z + 1 \\ -\frac{1}{2}x + \frac{1}{2} \\ \frac{1}{2}y + \frac{1}{2} \end{pmatrix} \quad H_5 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x + \frac{1}{2} \\ \frac{1}{2}y + \frac{1}{2} \\ \frac{1}{2}z + \frac{1}{2} \end{pmatrix}
\]

\[
H_6 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}z + \frac{1}{2} \\ \frac{1}{2}y + \frac{1}{2} \\ -\frac{1}{2}x + 1 \end{pmatrix} \quad H_7 \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x + 0 \\ -\frac{1}{2}z + \frac{1}{2} \\ -\frac{1}{2}y + 1 \end{pmatrix}
\]

⇒ leads to algorithm analog to 2D Hilbert and 2D Peano
⇒ uses only one pattern; each in only one orientation
3D Hilbert Curves – Variants

Different approximating polygons:

- same basic pattern:
  - same order of the eight sub-cubes
- differences only noticeable from the 2nd iteration
Different orientation of the sub-cubes:

- same basic pattern Grundmotiv, same approximating polygon
- differences only visible from 2nd iteration
3D Peano Curves

- Concentration on “serpentine” Peano curves (no Meander-type)
- still lots of different variants
- especially interesting are dimension-recursive variants:

  in each 3D cut, the sub-cubes are again traversed in Peano order
Parallelisation using Space-filling Curves

Problem setting:
- “mesh” (2D, 3D, ...) of \( N \) unknowns (\( N \gg 1000 \))
- solve linear system(s) of equations (maybe repeatedly with varying right-hand side)
- in the system, only spatially neighbouring unknowns are coupled

Parallelisation:
Distribute \( N \) unknowns to \( p \) partitions, such that
- each partition contains the same number of unknowns (load balancing)
- for as many unknowns as possible, all neighbours are in the same partition (\( \Rightarrow \) avoids communication between partitions)
Parallelisation using Space-filling Curves (2)

Further demand: adaptivity

- add further unknowns (during/depending on intermediate results) or drop unknowns
- (re-)partitioning required to be **fast**: must not cost more computation time than going on with a bad load balance
- “shape preserving”: if only few unknowns are added or dropped, the shape of partitions should not change strongly
  ⇒ only few unknowns then need to migrate to another partition

⇒ popular strategy: use **space-filling curves**
Hölder Continuity of Space-filling Curves

Definition: (Hölder continuous)

A function $f$ is called Hölder continuous with exponent $r$ on the interval $I$, if a constant $C > 0$ exists, such that for all $x, y \in I$:

$$\|f(x) - f(y)\|_2 \leq C |x - y|^r$$

Importance for space-filling curves:

- $|x - y|$ is the distance of the indices
- $\|f(x) - f(y)\|$ is the distance of the image points (in “space”)
- To prove: the Hilbert curve is Hölder continuous with exponent $r = d^{-1}$, where $d$ is the dimension:

$$\|f(x) - f(y)\|_2 \leq C |x - y|^{1/d} = C \sqrt[1/d]{|x - y|}$$
Hölder Continuity of the 3D Hilbert Curve

Proof analogous to simple continuity proof:

- given \( x, y \in I \); find an \( n \), such that \( 8^{-(n+1)} < |x - y| < 8^{-n} \)
- \( 8^{-n} \) is the interval length for the \( n \)-th iteration
  \( \Rightarrow [x, y] \) covers at most two neighbouring(!) intervals.

- per construction of the 3D Hilbert curve, the function values \( h(x) \) and \( h(y) \) are in two adjacent cubes of side length \( 2^{-n} \).
- the length of the space diagonal through the two adjacent cubes is \( 2^{-n} \cdot \sqrt{1^2 + 1^2 + 2^2} = 2^{-n} \cdot \sqrt{6} \), hence:

\[
\| h(x) - h(y) \|_2 \leq 2^{-n} \sqrt{6} = (8^{-n})^{1/3} \sqrt{6} = \left(8^{-(n+1)}\right)^{1/3} 8^{1/3} \sqrt{6} \\
\leq 2\sqrt{6} \cdot |x - y|^{1/3} \quad \text{q.e.d.}
\]
Hölder Continuity and Parallelisation

- for the Hilbert curve (also Peano curve and all contiguous, recursive SFC), we have:

\[ \| f(x) - f(y) \|_2 \leq C \sqrt{|x - y|} \]

- relates the distance \( |x - y| \) between indices to the distance \( \| f(x) - f(y) \| \) of (mesh) points

- gives relation between volume (number of grid cells/points) and extent (e.g. radius) of a partition

⇒ Hölder continuity gives a quantitative estimate for \textbf{compactness} of partitions