1.8

SPACE-FILLING CURVES AND A MEASURE OF COHERENCE

Douglas Voorhies
Hewlett Packard Co.
Chelmsford, Massachusetts

Traversal

Performance of ray tracers and other rendering algorithms can be improved by altering the primary ray generation sequence. The key is to better exploit coherence.

Coherence provides opportunities to use memory more effectively. Modern memory hierarchies employ caches, translation look-aside buffers, and paging, which depend upon extreme repetition in the reference pattern. In this environment, rendering approaches that minimize churning of the working set runs considerably faster, even if the total computation is the same.

Object–space rendering (e.g., ray tracing or reverse-mapped volumetric) typically processes the screen–space pixels in scanline order, which is mediocre in its churning of the working set. An object whose screen projection is several pixels across will have several consecutive pixels that enjoy the same intersections and similar lighting experiences, but the scanline traversal sequence then leaves the object and does not return for a long time. Thus, scanline traversal exploits only one dimension of object coherence.

An ideal traversal sequence visits all the pixels in the entire area of each object before moving on to the next object, thus exploiting coherence in two dimensions. Unfortunately, there is a wide range of on-screen object sizes, and neither their size nor position is known *a priori*. Furthermore, different levels of memory hierarchies work best with repetition within working sets of different sizes. Thus, the concept of *object*

ISBN 0-12-064480-0

area in this context may be extended to mean repetitious access at many size levels.

Two important properties for a traversal sequence which exploits coherence are that:

1. *All the pixels* in an area are visited before moving on.
2. The sequentially visited areas always are *adjacent*.

Both properties decrease the reloading of memory by minimizing the number of times an object area is exited and reentered. Assuming that the traversal sequence is fixed rather than adaptive, its selection must be based on shape alone.

Space-Filling Curves

Some space-filling curves can exploit coherence in two dimensions despite a range of object area sizes. As fractals, they are self-similar at multiple resolutions. Thus, if they do a good job of visiting all the pixels corresponding to a particular object area at one scale, then they may have

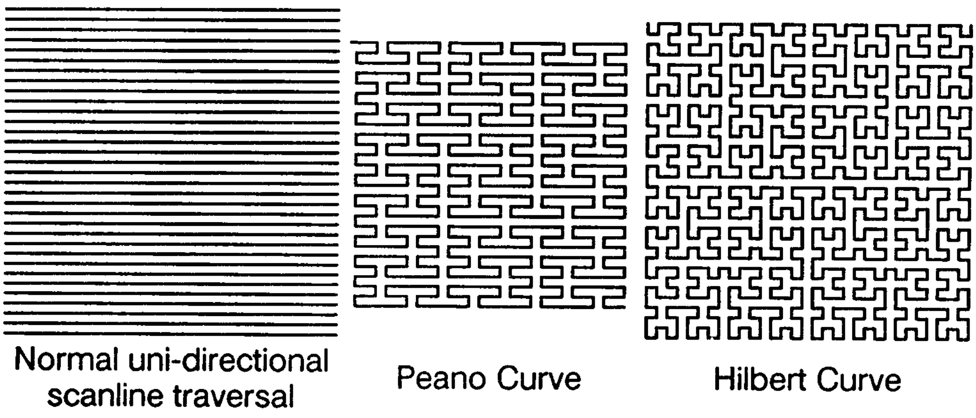

Figure 1.

this advantage for much larger or smaller object areas. As topologically continuous curves, areas visited always are adjacent. Let us look at two of them, as shown in Fig. 1.

Peano curves are recursive boustrophedonic patterns. They snake back and forth at multiple recursion levels, filling the plane. Since each pattern enters and exits from opposite rectangle corners, they implicitly use serpentines with an odd number of swaths, a minimum of three. Thus, they can be self-similar at 3×3, 9×9, 27×27, 81×81, etc. resolutions. (If squeezed in one dimension only, they can repeat as 2×3, 6×9, 18×27, 54×81, etc.)

A *Hilbert curve* is folded even more tightly. It can visit all the pixels in a 2×2, 4×4, 8×8, 16×16, etc. area before moving on, so its self-similar, area-filling patterns are more numerous and closer together in scale.

A Measure of Coherence

How can one measure the ability of a traversal sequence to exploit coherence, independent of a specific image and renderer? A pragmatic measure can be based on the notion of a memory working set. For a given object area, any traversal will consist of consecutive pixels that are within that object's screen projection and others that lie outside. Part of the working set of a renderer can be thought of as the *current object area*

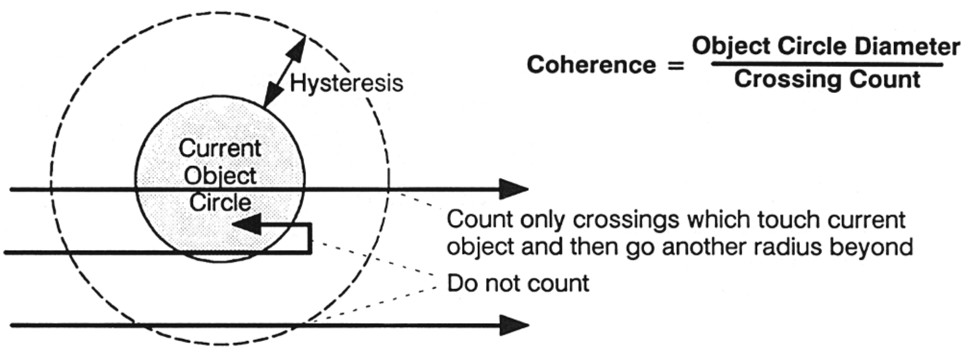

Figure 2.

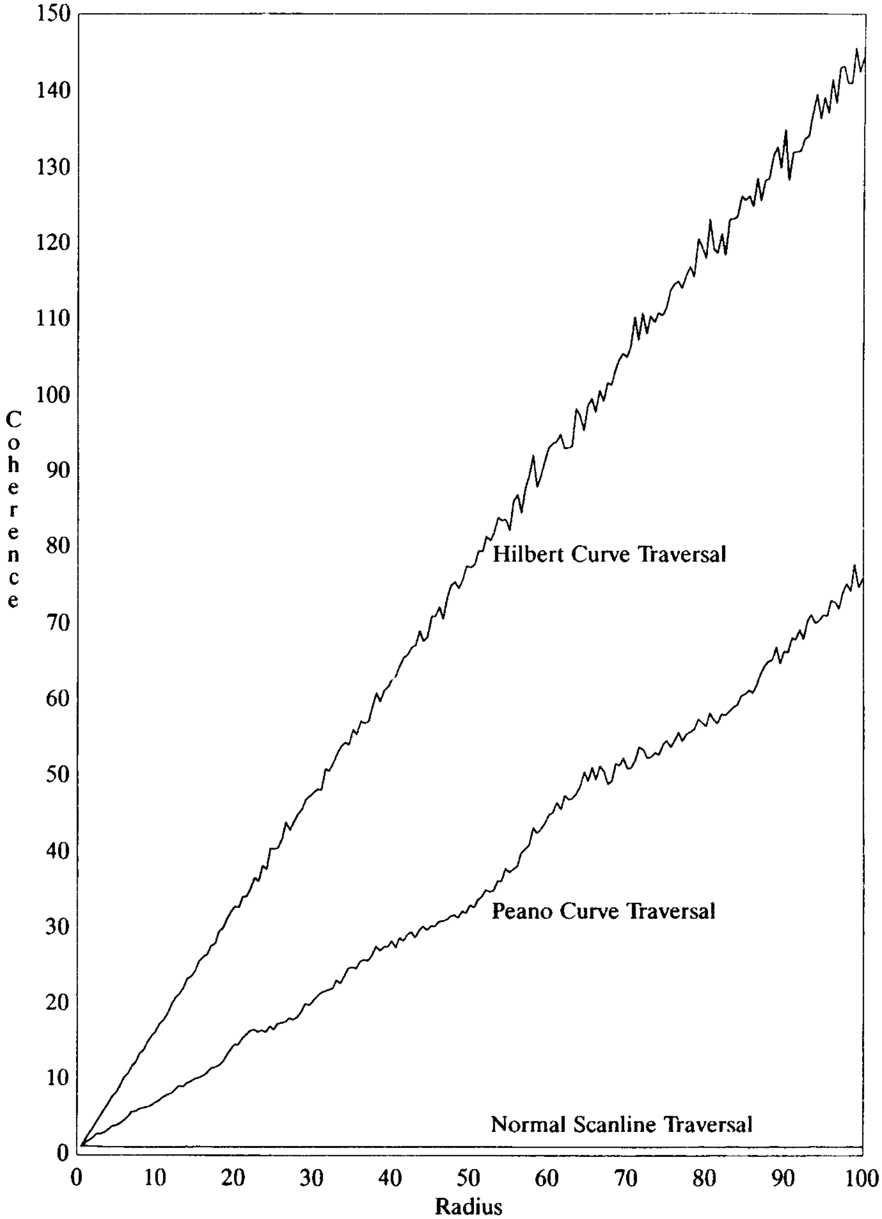

Figure 3. Average coherence of Peano and Hilbert traversal sequences for object radii of .5 to 100.

plus the recently touched ones. In this context, *high coherence* means staying within the current object area (or nearby ones) most of the time, and venturing outside rarely, until all the current object area pixels have been visited.

To make this concrete, I assume that an imaginary object projects as a circle in screen space. Choosing a circle favors no traversal direction over another. Coherence of a traversal sequence then is the current object circle's diameter divided by the number of times the sequence exits the circle. Hysteresis is added by only counting paths that exit the object circle and then venture another radius away; this avoids penalizing small excursions beyond the circle that immediately return. See Fig. 2.

By this pragmatic measure, object areas not touching the screen edges have a coherence measure of 1.00 for normal scanline traversal, since all sequences that penetrate the circle exit and continue all the way to the screen edge and thus are counted. For Fig. 3, a 1024×1024 screen is traversed by three methods: normal scanline traversal, a Peano curve, and a Hilbert curve. (Since the Peano curve did not fit evenly into 1024×1024, it was clipped.) The coherence of 1000 circles centered at random on-screen positions is averaged for a range of radii.

Both the Peano and Hilbert curves far exceed the *coherence = 1.00* measure of conventional scanline traversal. The Hilbert curve appears superior, and is easier to generate. Better performance from a real renderer can be expected simply by choosing either of these alternative traversal sequences.

See also I.7 A Peano Curve Generation Algorithm, Ken Musgrave