Radial Sweep

The Radial Sweep algorithm is a contour tracing algorithm that has been explained in some of the literature. Unlike its fancy name, the idea behind it is very simple. As a matter of fact, it turns out that the Radial Sweep algorithm is **identical** to Moore-Neighbor Tracing. So you must be asking: "Why did we bother to mention it here?". Well...

Moore-Neighbor tracing searches the Moore neighborhood of the current boundary pixel in a certain direction (we've chosen clockwise), until it finds a black pixel. It then declares that pixel as the current boundary pixel and proceeds as before.
The Radial Sweep algorithm does the exact same thing. On the other hand, it provides an interesting method for finding the next black pixel in the Moore neighborhood of a given boundary pixel.
The idea behind that method is the following:
every time you locate a new boundary pixel, make it your current pixel, **P,** and draw an **imaginary line segment** joining **P** to the **previous** boundary pixel. Then, **rotate** the segment about **P** in a clockwise direction until it hits a black pixel in **P**'s Moore neighborhood. Rotating the segment is identical to checking each pixel in the Moore neighborhood of **P**.

We have provided the following animated demonstration in order to explain how the Radial Sweep algorithm works and how similar it is to Moore-Neighbor tracing.

Radial Sweep

Demonstration

So... When does the Radial Sweep algorithm terminate?
Let's examine the behavior of the algorithm when the following stopping criteria are used...

Stopping Criterion 1

Let the Radial Sweep algorithm terminate when it visits the **start** pixel for a second time.
The following is an animated demonstration explaining why it would be a good idea to change that stopping criterion.

Demonstration:

A reason to change the stopping criterion

A point worth mentioning is that the performance of the Radial Sweep algorithm is identical to that of Moore-Neighbor tracing when this stopping criterion is used in both.

In the Square Tracing algorithm and Moore-Neighbor tracing, we found that using ***Jacob's stopping criterion*** (proposed by Jacob Eliosoff) greatly improved both algorithms' performance.
Jacob's stopping criterion requires that the algorithm terminates when it visits the **start** pixel for a second time in the same direction it did the first time around.
Unfortunately, we won't be able to use Jacob's stopping criterion in the Radial Sweep algorithm. The reason for this is the fact that the Radial Sweep algorithm doesn't define the concept of **the "direction" in which it enters a boundary pixel**. In other words, it's not clear (nor is it trivial to define) the "direction" in which a boundary pixel is entered in the algorithm.
Therefore, we will suggest another stopping criterion which doesn't depend on the direction in which you enter a certain pixel and will improve the performance of the Radial Sweep algorithm...

Stopping Criterion 2

Assume that each time a new boundary pixel, $\mathbf{P_i}$, is found by the algorithm, it is inserted in the **sequence** of boundary pixels as such: $\mathbf{P_1, P_2, P_3, ..., P_i}$; and is declared as the current boundary pixel.
(Assume $\mathbf{P_1}$ is the **start** pixel).
This means that we know the previous boundary pixel, $\mathbf{P_{i-1}}$, of every current boundary pixel, $\mathbf{P_i}$.
(As for the **start** pixel, we will assume that $\mathbf{P_0}$ is an imaginary pixel -not equivalent to ANY pixel on the grid- which comes before the **start** pixel in the sequence of boundary pixels).

With the above assumptions in mind, we can define our stopping criterion:
The algorithm terminates when:
a) the current boundary pixel, $\mathbf{P_i}$, has appeared **previously** as pixel $\mathbf{P_j}$ (where $\mathbf{j < i}$) in the sequence of boundary pixels,
AND
b) $\mathbf{P_{i-1} = P_{j-1}}$.

In other words, the algorithm terminates when it visits a boundary pixel, P, for a **second** time provided that the boundary pixel before P (in the sequence of boundary pixels) the second time around, is the **same** pixel which was before P when P was **first** visited.

If this stopping criterion was satisfied and the algorithm didn't terminate, the Radial Sweep algorithm will proceed to trace the **same** boundary for a second time.

The performance of the Radial Sweep algorithm using this stopping criterion is similar to the perfomance of Moore-Neighbor tracing using Jacob's stopping criterion.

All comments and questions are welcomed...
abeer.ghuneim@mail.mcgill.ca