Bresenham's Circle Drawing Algorithm

David Oniani
onianidavid@luther.edu

April 2, 2021

1 Derivation

Let the radius of the circle be given and equal r. Any point (x, y) on the circle satisfies the following:

$$x^2 + y^2 = r^2 \tag{1}$$

We consider the upper 45 degrees in the first quadrant of the circle. We can place the next point either to the east $(x + 1, y)$ or southeast $(x + 1, y - 1)$. We define errors for both of these points:

$$e(x + 1, y) = (x + 1)^2 + y^2 - r^2 \tag{2}$$
$$e(x + 1, y - 1) = (x + 1)^2 + (y - 1)^2 - r^2 \tag{3}$$

We now define the decision parameter d:

$$d = e(x + 1, y) + e(x + 1, y - 1) = 2(x + 1)^2 + y^2 + (y - 1)^2 - 2r^2 \tag{4}$$

Since the x coordinate always increases, the next point is $(x + 1, y_n)$ and we have:

$$d_n = 2(x + 2)^2 + {y_n}^2 + (y_n - 1)^2 - 2r^2 \tag{5}$$

Now, let us calculate the difference:

$$d_n - d = 2(x + 2)^2 + {y_n}^2 + (y_n - 1)^2 - 2r^2 - (2(x + 1)^2 + y^2 + (y - 1)^2 - 2r^2) \tag{6}$$
$$d_n - d = 2(2x + 3) + \left({y_n}^2 - y^2\right) + \left((y_n - 1)^2 - (y - 1)^2\right) \tag{7}$$

Finally, we get:

$$d_n - d = 2(2x + 3) \implies d_n = d + 4x + 6 \quad \text{if } d \leq 0 \ (y_n = y) \tag{8}$$
$$d_n - d = 2(2x + 3) - 4y + 4 \implies d_n = d + 4(x - y) + 10 \quad \text{if } d > 0 \ (y_n = y - 1) \tag{9}$$
$$d_0 = 2(0 + 1)^2 + r^2 + (r - 1)^2 - 2r^2 \implies d_0 = 3 - 2r \quad \text{at initial point } (0, r) \tag{10}$$

2 Implementation

- Time Complexity: $O(r)$ where r is the radius of the circle
- Space Complexity: $O(1)$

```python
import matplotlib.pyplot as plt


def bresenham(radius: float) -> None:
    """Draw a circle using Bresenham's Circle Drawing Algorithm."""

    # Initial parameters
    x: int = 0
    y: float = radius
    d: float = 3 - 2 * radius

    # Only consider upper 45 degrees in the first quadrant and draw a circle
    while x <= y:
        # Plot eight points using eight-way symmetry
        plt.scatter([x, x, -x, -x, y, y, -y, -y], [y, -y, y, -y, x, -x, x, -x])

        # Make decisions and update parameters
        if d <= 0:
            d += 4 * x + 6
        else:
            d += 4 * (x - y) + 10
            y -= 1
        x += 1

    # Show the drawing
    plt.title("Circle Drawing Using Bresenham's Circle Drawing Algorithm")
    plt.show()
```

A Python Implementation of Bresenham's Circle Drawing Algorithm