



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE RORAIMA**  
CENTRO DE CIÊNCIAS E TECNOLOGIAS  
DEP. DE CIENCIA DA COMPUTACAO  
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)  
Prof. Dr. Luciano Ferreira Silva



**EDUARDO HENRIQUE DE ALMEIDA IZIDORIO**

**RASTERIZAÇÃO DE CIRCUNFERÊNCIAS**

**BOA VISTA, RR**  
**FEVEREIRO 2025**



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE RORAIMA**  
CENTRO DE CIÊNCIAS E TECNOLOGIAS  
DEP. DE CIÊNCIA DA COMPUTAÇÃO  
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)  
Prof. Dr. Luciano Ferreira Silva



**EDUARDO HENRIQUE DE ALMEIDA IZIDORIO**

## **RASTERIZAÇÃO DE CIRCUNFERÊNCIAS**

Relatório de pesquisa apresentado ao Centro de  
Ciência da Computação do Curso de Ciência da Computação  
da Universidade Federal de Roraima, como requisito parcial  
para obtenção de notas da Disciplina COMPUTAÇÃO GRÁFICA  
– DCC 703, sob orientação do professor Dr. Luciano F. Silva

**BOA VISTA, RR**  
**FEVEREIRO 2025**



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE RORAIMA**  
CENTRO DE CIÊNCIAS E TECNOLOGIAS  
DEP. DE CIENCIA DA COMPUTACAO  
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)  
Prof. Dr. Luciano Ferreira Silva



## SUMÁRIO

|   |    |
|---|----|
| 1. JUSTIFICATIVA.....                     | 4  |
| 2. INTRODUÇÃO .....                       | 4  |
| 3. OBJETIVOS .....                        | 5  |
| 4. ALGORITMO DA EQUAÇÃO PARAMÉTRICA ..... | 5  |
| 5. ALGORITMO INCREMENTAL .....            | 7  |
| 6. ALGORITMO DE BRESENHAM .....           | 7  |
| 7. COMPARAÇÃO DOS ALGORITMOS .....        | 8  |
| 8. CONCLUSÃO .....                        | 9  |
| 9. ANEXOS .....                           | 10 |



## 1. Justificativa

Este relatório foi concebido com o objetivo de documentar o desenvolvimento de uma aplicação que exemplifica o funcionamento de três algoritmos de rasterização de circunferência diferentes, o algoritmo de Equação paramétrica, incremental com simetria e o algoritmo de Bresenham. Neste documento serão mostradas análises, comparações e conclusões acerca dos supracitados algoritmos.

## 2. Introdução

A rasterização de circunferências é um processo fundamental na computação gráfica, sendo responsável por transformar representações matemáticas de curvas em sequências de pixels exibidas em dispositivos de saída. Esse procedimento é amplamente utilizado em diversas aplicações, como simulações gráficas, modelagem de objetos e renderização de cenas em jogos digitais.

Diferentes algoritmos podem ser utilizados para desenhar um círculo na tela, cada um apresentando características específicas em termos de precisão, desempenho e complexidade computacional. Neste trabalho, são explorados três algoritmos principais para rasterização de circunferências:

- **Algoritmo Incremental:** Baseia-se em aproximações trigonométricas para calcular os pontos da circunferência de forma incremental. Embora sua implementação seja relativamente simples, pode apresentar pequenas imprecisões devido ao acúmulo de erros de arredondamento.
- **Algoritmo de Bresenham:** Uma abordagem eficiente baseada em aritmética inteira, que evita cálculos com números de ponto flutuante. Esse método minimiza os custos computacionais e garante um traçado preciso, tornando-se uma escolha popular para a rasterização de círculos.



- **Equação Paramétrica:** Utiliza funções trigonométricas para calcular diretamente os pontos da circunferência, proporcionando alta precisão na renderização. No entanto, a necessidade de cálculos com seno e cosseno pode impactar seu desempenho em sistemas com recursos limitados.

Cada um desses algoritmos apresenta vantagens e desvantagens que impactam diretamente sua aplicação em diferentes contextos. A escolha do método mais adequado depende das necessidades específicas da aplicação, equilibrando fatores como desempenho, precisão e complexidade de implementação.

### 3. Objetivos

Este trabalho tem como principais objetivos:

- **Implementar os algoritmos de rasterização de circunferências:** Desenvolver um programa que desenhe circunferências utilizando os métodos **Incremental**, **Bresenham** e **Equação Paramétrica**.
- **Comparar os três métodos:** Avaliar a eficiência, precisão e desempenho computacional de cada algoritmo, identificando suas vantagens e limitações.
- **Analisar a aplicabilidade dos algoritmos:** Compreender o impacto de cada abordagem em contextos práticos, considerando fatores como velocidade de execução e qualidade da renderização.

A implementação e a análise desses algoritmos permitirão uma compreensão mais aprofundada sobre técnicas de rasterização, auxiliando na escolha da melhor estratégia para diferentes aplicações gráficas.

### 4. Algoritmo de Equação Paramétrica

#### Definição

O algoritmo de rasterização de circunferências baseado na **equação paramétrica** calcula os pontos de uma circunferência a partir de suas equações trigonométricas:

$$x = x_c + r \cos(\theta)$$

$$y = y_c + r \sin(\theta)$$



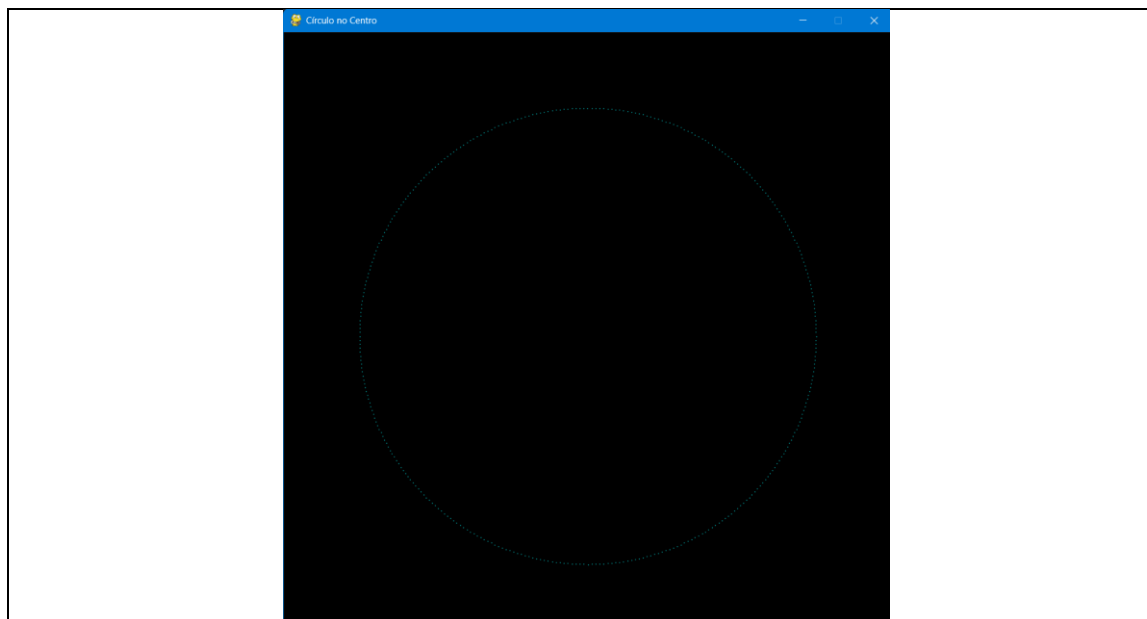
Onde  $X_c$ ,  $Y_c$  são as coordenadas do centro da circunferência,  $r$  é o raio, e  $\theta$  é o ângulo variando de 0 a 360 graus.

Cada ponto é arredondado para a grade de pixels mais próxima e desenhado na tela.

### **Análise:**

Esse algoritmo é simples e direto, garantindo precisão matemática na definição dos pontos da circunferência. No entanto, seu desempenho é afetado pelo uso de funções trigonométricas, que são computacionalmente custosas. Além disso, a necessidade de arredondamento pode introduzir pequenas imprecisões na exibição.

### **Execução em código:**



## **5. Algoritmo de Rasterização Incremental**

### **Definição**

O algoritmo incremental utiliza aproximações trigonométricas para calcular novos pontos da circunferência a partir de um valor inicial. Em vez de calcular



diretamente com seno e cosseno, ele utiliza fórmulas de incremento baseadas em pequenas variações angulares:

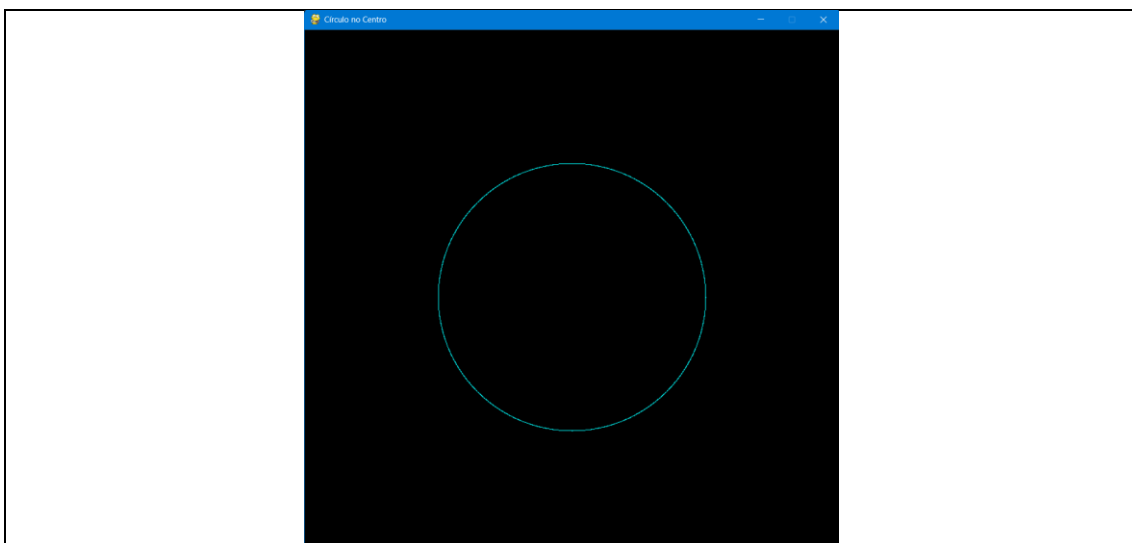
$$\begin{aligned}x' &= x \cos(\Delta\theta) - y \sin(\Delta\theta) \\y' &= y \cos(\Delta\theta) + x \sin(\Delta\theta)\end{aligned}$$

onde  $\Delta\theta = \frac{1}{r}$ , reduzindo a necessidade de cálculos complexos a cada iteração.

### Análise

Esse método é mais eficiente que a equação paramétrica porque reduz a quantidade de operações trigonométricas. No entanto, a precisão pode ser afetada pelo acúmulo de erros de arredondamento ao longo do desenho da circunferência.

### Execução em código:





## 6. Algoritmo de Bresenham para Circunferências

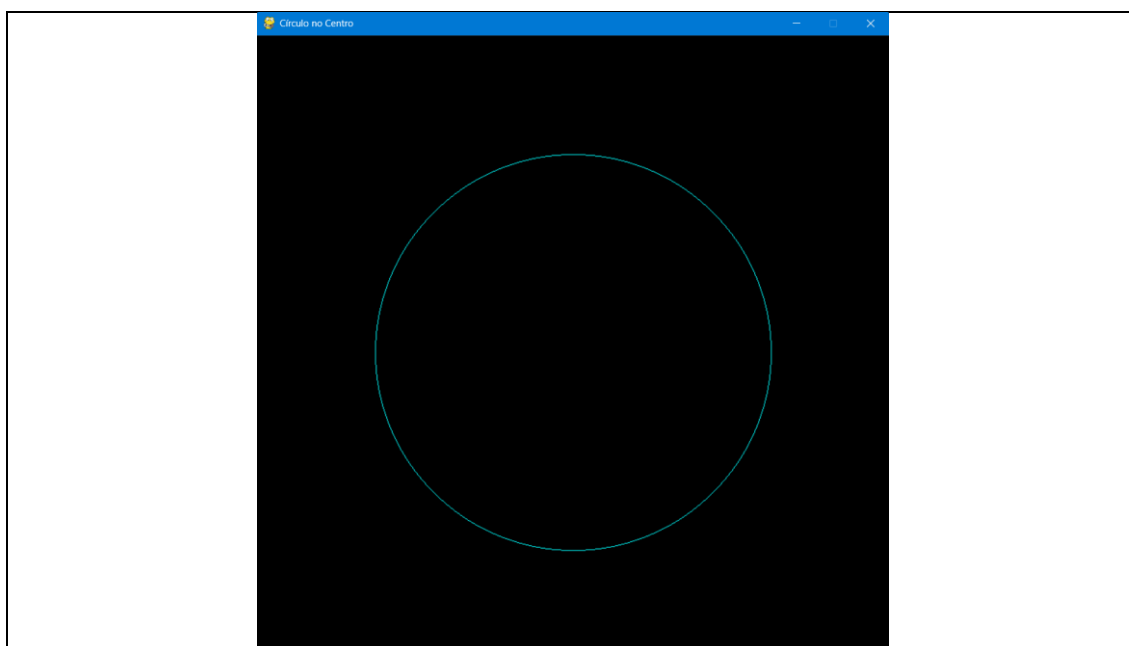
### Definição

O algoritmo de Bresenham é uma abordagem baseada em aritmética inteira, que evita cálculos com números de ponto flutuante. Ele utiliza um critério de decisão baseado na equação da circunferência para determinar quais pontos devem ser ativados na grade de pixels.

### Análise

Este é um dos algoritmos mais eficientes para rasterização de circunferências, pois elimina cálculos trigonométricos e trabalha apenas com somas e subtrações. Além disso, ele utiliza simetria octogonal, reduzindo ainda mais a quantidade de cálculos necessários.

### Execução em código:







## 7. Comparação dos Algoritmos

| Algoritmo           | Precisão | Desempenho | Uso de Trigonometria | Aplicação Ideal                          |
|---------------------|----------|------------|----------------------|--|
| Equação Paramétrica | Alta     | Baixo      | Sim                  | Cálculo exato, gráficos de alta precisão |
| Incremental         | Média    | Médio      | Parcialmente         | Simulações e gráficos interativos        |
| Bresenham           | Boa      | Alto       | Não                  | Aplicações em tempo real, como jogos     |

O **algoritmo de Bresenham** é a melhor escolha para aplicações que exigem alto desempenho, enquanto a **equação paramétrica** é ideal para situações que demandam alta precisão, mas não são sensíveis a tempo de execução.

## 8. Conclusão

A escolha do algoritmo ideal para rasterização de circunferências depende do contexto da aplicação. O algoritmo de equação paramétrica é simples e preciso, mas pode ser computacionalmente caro devido ao uso de funções trigonométricas. O método incremental melhora a eficiência ao reduzir o número de cálculos trigonométricos, mas pode sofrer com acúmulo de erros de arredondamento. Por fim, o algoritmo de Bresenham se destaca como a opção mais eficiente, sendo amplamente utilizado em aplicações gráficas em tempo real.

Dessa forma, em sistemas que priorizam desempenho, o algoritmo de Bresenham é a melhor escolha. Já para aplicações que necessitam de alta precisão matemática, a equação paramétrica continua sendo uma alternativa viável.



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE RORAIMA**  
CENTRO DE CIÊNCIAS E TECNOLOGIAS  
DEP. DE CIENCIA DA COMPUTACAO  
DCC703 – COMPUTAÇÃO GRÁFICA (2024.2)  
Prof. Dr. Luciano Ferreira Silva



## **9. ANEXOS (Link do Github)**

[Github – Computer Graphics](#)