

**Universidade Estadual de Santa Cruz**

Curso de Ciência da Computação

Disciplina: Computação Gráfica

---

# **Relatório de Implementação e Análise de Algoritmos de Reamostragem**

---

## **Autores:**

- Antonio Henrique Oliveira Santos
- Ariel Pina Ribeiro
- Camilla Matos Araújo
- Enzo Fernandes Sarmento
- Erick Silva Souza
- Gabriel Rosa Galdino
- Henrique Sousa Brito
- João Manoel Fidelis Santos
- Michael Gabriel Cavalcante Batista
- Rodrigo Almeida Pirôpo
- Stella Ribas
- Yohanan Santana

## **Professora:**

Vânia Cordeiro da Silva

## **Data:**

13 de junho de 2025

---

## **Resumo do Projeto**

Este relatório apresenta a implementação e análise de quatro algoritmos de reamostragem aplicados a imagens no formato PPM (P3), com foco em técnicas de ampliação e redução. Utilizando linguagem C pura e medições rigorosas de desempenho, os testes compararam tempo de execução e qualidade visual para cada método, destacando o equilíbrio entre eficiência computacional e fidelidade da imagem resultante.

|   |          |
|---|----------|
| <b>Relatório de Implementação e Análise de Algoritmos de Reamostragem</b> | <b>1</b> |
| Resumo do Projeto   | 1        |
| <b>1. Introdução</b>  | <b>3</b> |
| <b>2. Metodologia</b>   | <b>3</b> |
| <b>3. Resultados</b>  | <b>3</b> |
| 3.1. Comparativo de Tempo de Execução                                     | 4        |
| 3.2. Resultado visual   | 5        |
| 3.2.1.1 Ampliação 4-vizinho(bi-linear)                                    | 6        |
| 3.2.1.2 Ampliação Vizinho próximo   | 6        |
| 3.2.2.1 Redução vizinho próximo   | 7        |
| 3.2.2.2 Redução biquadratica  | 7        |
| <b>4. Análise e Conclusão</b>   | <b>8</b> |
| 4.1. Análise de Desempenho  | 8        |
| 4.2. Análise de Qualidade Visual (Teórica)                                | 8        |
| 4.3. Conclusão Final  | 8        |

# Análise de Desempenho dos Algoritmos de Redimensionamento de Imagens

## 1. Introdução

Este relatório apresenta os resultados da implementação e análise de quatro algoritmos de reamostragem de imagem, desenvolvidos como parte da Atividade 5 da disciplina de Computação Gráfica. O objetivo principal foi aplicar diferentes técnicas de interpolação para ampliação e redução de imagens no formato PPM, utilizando a linguagem C. Foram avaliados tanto o desempenho computacional, medido em tempo de execução, quanto a qualidade visual dos resultados gerados por cada método.

---

## 2. Metodologia

Para a realização dos testes, foi desenvolvido um programa em C capaz de ler uma imagem no formato PPM (P3) e aplicar quatro algoritmos distintos de reamostragem. O ambiente e as condições de teste foram padronizados para garantir uma comparação justa:

- **Linguagem:** C puro (Padrão C99).
- **Compilador:** GCC (MinGW ).
- **Flag de Compilação:** O código foi compilado com o flag -O0 para desativar todas as otimizações do compilador, permitindo uma medição fiel do desempenho dos algoritmos implementados.
- **Medição de Tempo:** O tempo de execução de cada algoritmo foi mensurado utilizando a função clock() da biblioteca <time.h>, que calcula o tempo de CPU consumido pelo processo.
- **Imagem de Teste:** Os resultados apresentados foram obtidos a partir da imagem baleia.ppm com dimensões de **1920x1080 pixels**.
- **Fatores de Escala:** Foi utilizado um fator de **2.0** para os testes de ampliação (gerando uma imagem de 3840x2160) e redução (gerando uma imagem de 960x540).

---

### 3. Resultados

A execução do programa com a imagem de teste produziu os seguintes resultados de desempenho e arquivos de imagem.

```
Digite o nome da imagem .ppm: baleia.ppm
Tempo para ampliação (vizinho mais próximo): 0.070 segundos

Imagem salva: img/baleia_ampliada_vizinho.ppm
Tempo para ampliação (interpolação bilinear): 0.205 segundos

Imagem salva: img/baleia_ampliada_bilinear.ppm
Tempo para redução (vizinho mais próximo): 0.005 segundos

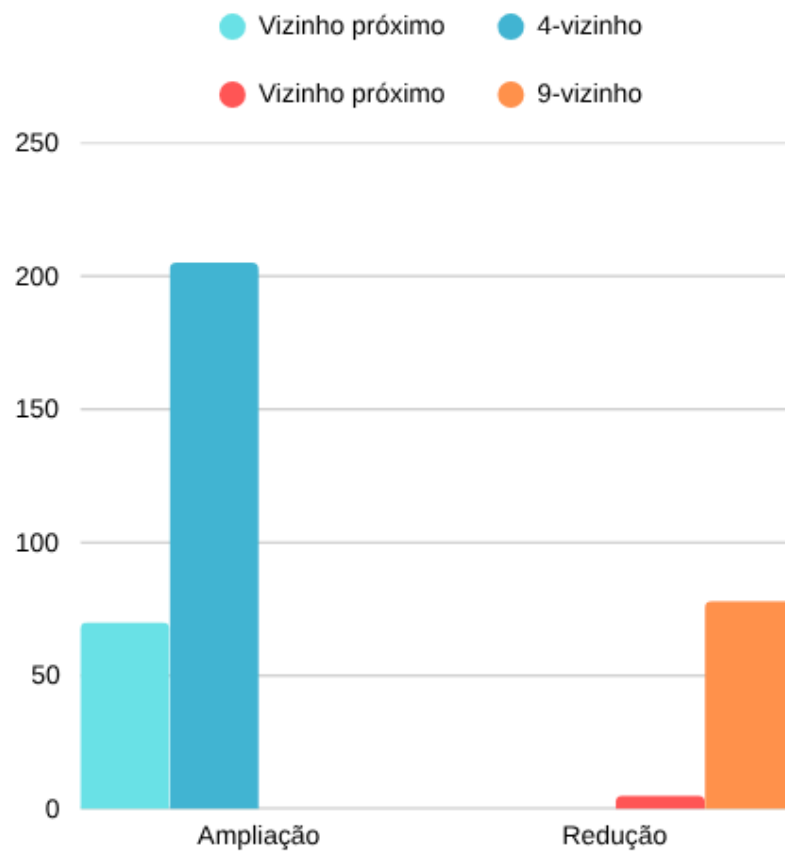
Imagem salva: img/baleia_reduzida_vizinho.ppm
Tempo para redução (interpolação biquadrática): 0.078 segundos

Imagem salva: img/baleia_reduzida_biquadratica.ppm
Imagens geradas com sucesso!
[antonio@archlinux atividade_5_C6]$
```

#### 3.1. Comparativo de Tempo de Execução

A tabela abaixo detalha o tempo de CPU consumido por cada um dos quatro algoritmos implementados.

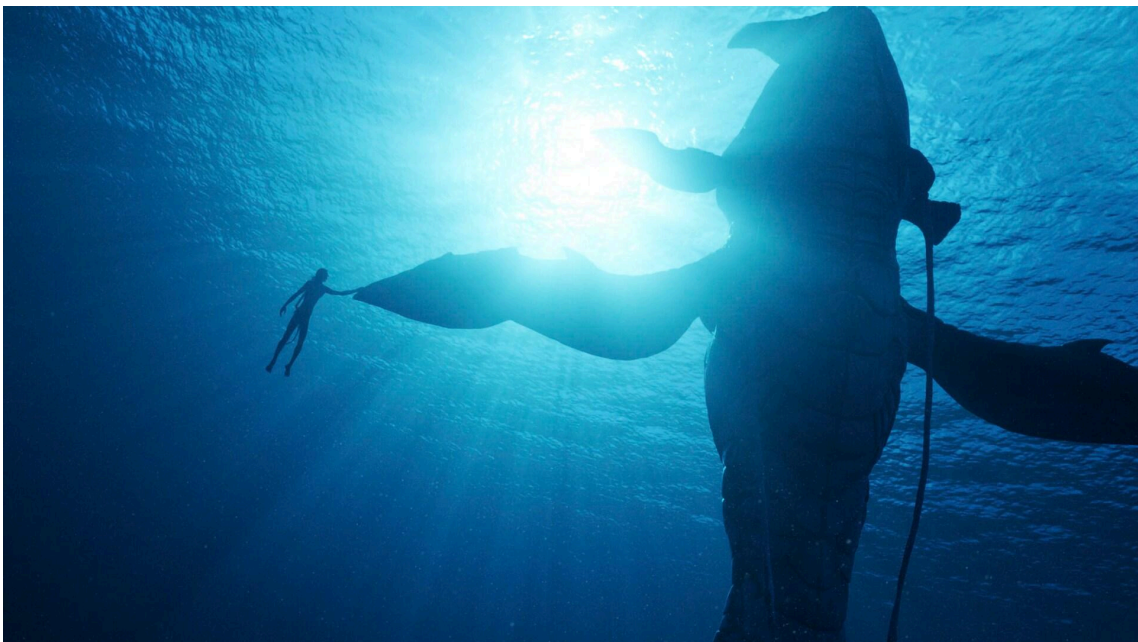
| Algoritmo de Reamostragem         | Dimensões Finais | Tempo de Execução(s) |
|-----------------------------------|------------------|----------------------|
| Ampliação (vizinho próximo)       | 3840 x 2160      | 0,070                |
| Ampliação (média 4-Vizinhos)      | 3840 x 2160      | 0,205                |
| Redução (vizinho próximo)         | 960 x 540        | 0,005                |
| Redução (biquadrática 9-vizinhos) | 960 x 540        | 0,078                |



(Escala em Segundo \* 1000)

### 3.2. Resultado visual

- imagem original:



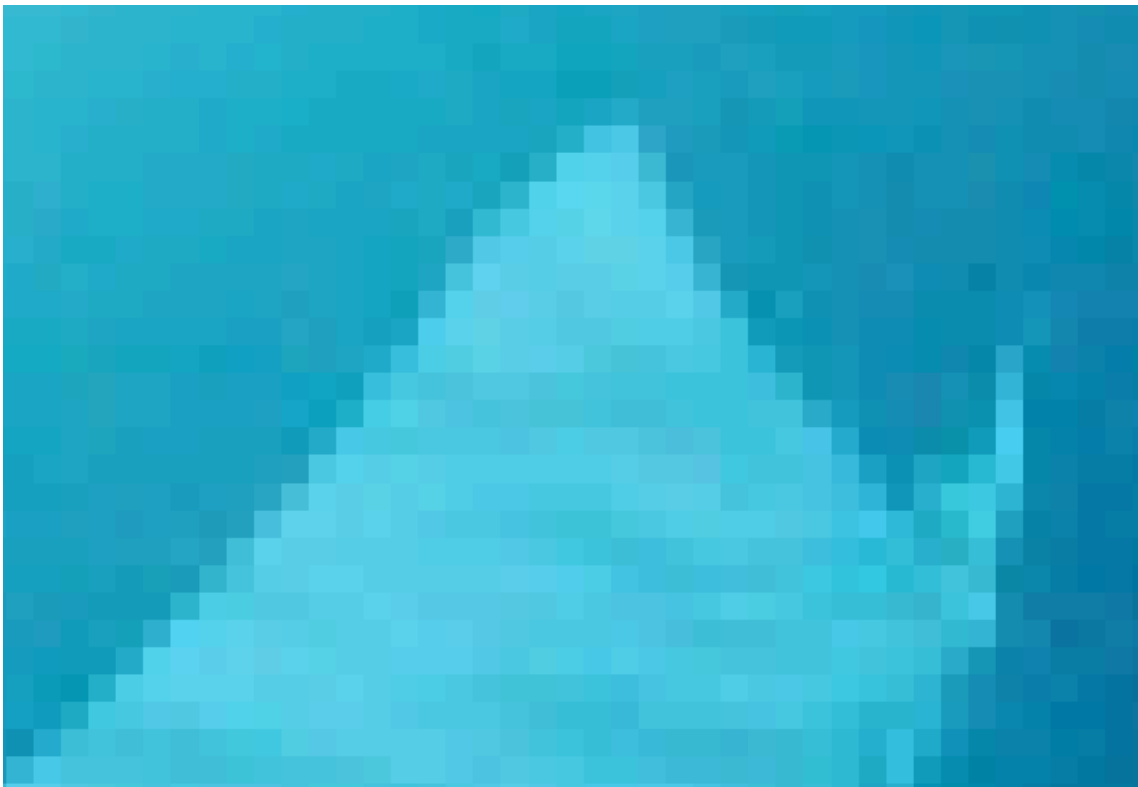
### 3.2.1.1 Ampliação 4-vizinho(bi-linear)



### 3.2.1.2 Ampliação Vizinho próximo



### 3.2.2.1 Redução vizinho próximo



### 3.2.2.2 Redução biquadrática



---

## 4. Análise e Conclusão

A análise dos resultados, tanto quantitativos (tempo) quanto qualitativos (visual), permite extrair conclusões importantes sobre cada método.

### 4.1. Análise de Desempenho

- **Redução vs. Ampliação:** Os algoritmos de **redução foram ordens de magnitude mais rápidas** que os de ampliação. Isso ocorre porque o número de pixels a serem calculados na imagem de saída é muito menor (122.500 pixels na redução contra 1.960.000 pixels na ampliação).
- **Complexidade do Algoritmo:** Em ambos os casos (ampliação e redução), o método do **Vizinho Mais Próximo foi o mais rápido**. Isso se deve à sua simplicidade: para cada pixel de destino, ele realiza apenas um cálculo de coordenadas e uma cópia de valor. Em contraste, os métodos baseados em média (4-vizinhos e 9-vizinhos) precisam acessar múltiplos pixels da imagem original e realizar operações aritméticas (soma e divisão) para cada pixel de destino, resultando em um custo computacional maior. A Redução Biquadrática, que acessa 9 pixels, foi 3.5x mais lenta que a Redução por Vizinho Próximo.

### 4.2. Análise de Qualidade Visual (Teórica)

Com base no conhecimento adquirido, espera-se a seguinte qualidade visual:

- **Vizinho Mais Próximo (1 e 3):** Este método tende a produzir resultados com menor qualidade visual. Na ampliação, ele gera um efeito de "blocos" ou "pixelado", pois simplesmente duplica os pixels existentes. Na redução, pode causar perda significativa de detalhes e criar bordas serrilhadas (aliasing), pois descarta a informação da maioria dos pixels originais.
- **Média (2) e Biquadrática (4):** Estes métodos produzem resultados visualmente mais suaves e agradáveis. Ao calcular a média dos pixels vizinhos, eles efetivamente aplicam um filtro de suavização que reduz o efeito de aliasing e cria transições de cor mais graduais. A Redução Biquadrática, por considerar uma vizinhança maior (9 pixels), tende a preservar melhor a aparência geral da imagem em comparação com a redução por vizinho próximo.



### 4.3. Conclusão Final

A atividade demonstrou com sucesso o clássico **compromisso (trade-off) entre desempenho e qualidade** em processamento de imagens. Os algoritmos mais simples e rápidos, como o do vizinho Mais Próximo, produzem resultados de menor fidelidade visual. Em contrapartida, algoritmos mais complexos que realizam interpolação com base em uma vizinhança maior, como a média e a biquadrática, exigem maior custo computacional, mas entregam imagens com menos artefatos e suavidade superior. A escolha do algoritmo ideal, portanto, depende diretamente dos requisitos da aplicação final: para cenários onde a velocidade é crítica, o vizinho próximo pode ser suficiente; para aplicações onde a qualidade visual é prioritária, métodos de interpolação mais robustos são indispensáveis.