

Universidade Federal do Maranhão

Departamento de Engenharia da Computação

Autor: Professor Doutor Thales Levi Azevedo Valente

Descrição: Projeto de Processamento de Imagens - Laboratório de Programação em Linguagem C

Data: 26-11-2023

Descrição:

Desenvolva um programa em C que realiza operações básicas de processamento de imagens, incluindo leitura, aplicação de filtros e gravação das imagens processadas. O projeto deve lidar com formatos de imagem populares como JPEG, PNG ou BMP, utilizando bibliotecas externas para simplificar o manuseio desses formatos.

Objetivos de Aprendizado:

Desenvolver competências práticas em programação na linguagem C, enfatizando leitura, processamento e escrita de imagens. Este projeto visa a utilização de bibliotecas externas e a implementação de algoritmos fundamentais de processamento de imagens.

- Ler e escrever arquivos de imagem em formatos comuns.
- Manipular dados de imagem utilizando vetores e matrizes.
- Aplicar conceitos de alocação dinâmica de memória.
- Implementar algoritmos de processamento de imagens.

Requisitos:

- Leitura e Escrita de Arquivos:
- Ler imagens em formatos como JPEG, PNG ou BMP.
- Gravar a imagem processada no mesmo formato.

Manipulação de Dados:

- Utilizar vetores e matrizes para manipular os pixels da imagem.
- Empregar alocação dinâmica de memória.
- Processamento de Imagens:

Implementar ao menos dois dos seguintes filtros:

- Escala de cinza.
- Filtro de média (blur).
- Detecção de bordas (Sobel).
- Inversão de cores (negativo).

Interface de Usuário:

- Criar um menu para escolher entre diferentes filtros.

Filtros e Matrizes:

1. Escala de Cinza
 - Conversão RGB para cinza: $\text{Gray} = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B$.

2. Filtro de Média (Blur):

- Kernel (matriz) de exemplo para blur 3x3:

1/9 1/9 1/9

1/9 1/9 1/9

1/9 1/9 1/9

3. Detecção de Borda (Sobel):

- Kernel horizontal (Gx):

-1 0 1

-2 0 2

-1 0 1

- Kernel vertical (Gy):

-1 -2 -1

0 0 0

1 2 1

4. Inversão de Cores (Negativo):

- Para cada pixel: Invertido = 255 - Original.

Bibliotecas Externas:

- stb_image.h e stb_image_write.h para leitura e escrita de imagens.
- Essas bibliotecas podem ser baixadas de <https://github.com/nothings/stb>.
- Elas são bibliotecas de cabeçalho únicas, o que significa que você só precisa incluir os arquivos .h no seu projeto (coloque na pasta do seu arquivo .c)

Etapas Sugeridas:

- Dia 1: Estrutura Básica, Leitura de Arquivos e Escrita de Arquivo.
- Dia 2: Alocação Dinâmica e Estrutura de Dados.
- Dia 3-5: Implementação de Filtros.
- Dia 6: Testes Finais.

Entrega (obs: tudo em .zip):

- Código-fonte do programa.
- Documentação explicativa.
- Exemplos de imagens processadas.

Avaliação:

- Corretude do código-fonte.
- Implementação dos requisitos.
- Qualidade da documentação.
- Eficácia dos filtros de imagem implementados.

Referências Úteis:

[Programação em C - Capítulo 1 \(inpe.br\)](#)

[Edge detection with Sobel filters \(jameshfisher.com\)](#)

[GitHub - petermlm/SobelFilter: Implementation of the Sobel Filter in C](#)

[Grayscale image - Rosetta Code](#)

[Projetos de PDI \(wenyaraujo.github.io\)](#)

[Edge detection with Sobel filters \(jameshfisher.com\)](#)

[Programação em C - Capítulo 1 \(inpe.br\)](#)

[An Implementation of Sobel Edge Detection - Rhea \(projectrhea.org\)](#)

[GitHub - nothings/stb: stb single-file public domain libraries for C/C++](#)