



INSTITUTO FEDERAL

Triângulo Mineiro
Campus Paracatu

**INSTITUTO FEDERAL DO TRIÂNGULO MINEIRO -
CAMPUS PARACATU**

**Curso: Tecnologia em Análise e Desenvolvimento
de Sistemas**

Disciplina: Programação de Computadores II

Professor: Josimar Viana

Trabalho Prático 01 (15 pontos)

Processamento de Imagens

O processamento digital de imagens usa intensivamente vetores e matrizes. Uma imagem é geralmente representada como uma matriz de pixels, onde cada pixel é descrito por um ou mais valores inteiros que indicam seus níveis de cor ou de luminosidade. Por isso, processar imagens é uma ótima forma de exercitar o uso de matrizes.

O formato PGM

Para facilitar a leitura e escrita dos arquivos de imagem, neste projeto será adotado o formato de imagem Portable Gray Map (PGM), um formato de imagem em níveis de cinza bem simples e fácil de ler/escrever. Boa parte dos programas de tratamento de imagens reconhece o formato PGM. Exemplo de imagem no formato PGM.

```
P2

# this is a PGM image

24 7

15

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Existem duas variantes do formato PGM: PGM plain (P2) e PGM raw (P5). O P2 deve ser implementado obrigatoriamente nesta atividade. A

implementação do P5 é opcional mas contará 3 pontos extras, se realizada. Mais detalhes sobre essas variantes podem ser obtidas no seguinte endereço: <http://netpbm.sourceforge.net/doc/pgm.html>

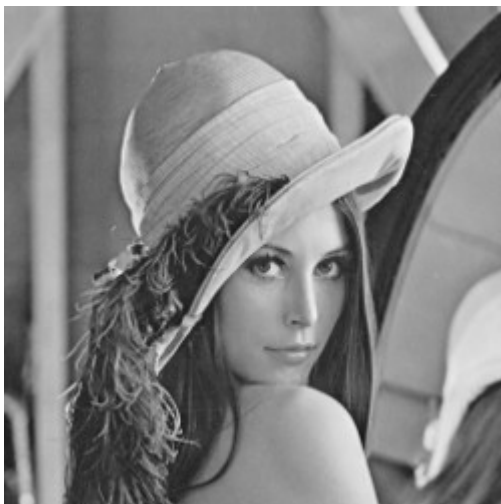
Filtros de Imagem

Filtro negativo

Este projeto visa construir filtros simples para imagens digitais em níveis de cinza, no formato PGM (P2 e P5) com pixels de 8 bits (1 byte).

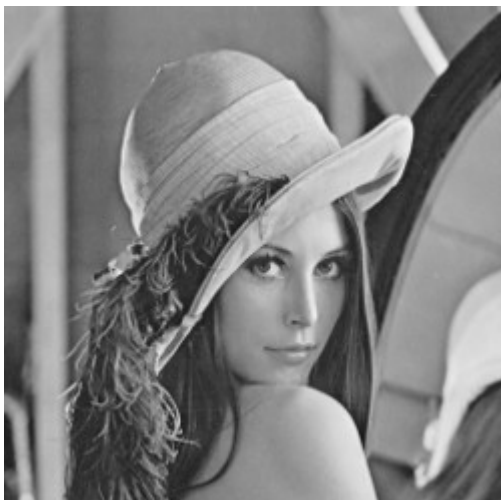
O filtro negativo consiste em converter cada pixel da imagem em seu complemento. Sendo max o valor máximo para um pixel na imagem, o complemento de um pixel com valor v seria $\text{max}-v$.

Exemplo de entrada e de saída:



Filtros de rotação simples

O filtro de rotação gira uma imagem em 90° no sentido horário. Exemplo de entrada e de saída:



O trabalho poderá ser realizado em grupo de até 3 (três) alunos. Não será aceito maior número de participantes por grupo.

Cada grupo deve implementar um dos filtros listados abaixo.

Filtros de rotação livre - Grupo 1

O filtro de rotação gira uma imagem em um ângulo θ (>0) no sentido horário, em relação ao seu centro (vide rotação de imagens 2D).

Requisitos:

O tamanho da imagem de saída deve ser ajustado para não cortar os cantos da imagem. Os espaços vazios gerados pela rotação devem ser preenchidos com a cor branca. O ângulo de rotação é informado pela opção -a; se não for informado, por default $\theta = 90^\circ$.



Filtros de limiar - Grupo 2

O filtro de limiar (threshold filter) converte uma imagem em tons de cinza em uma imagem em preto-e-branco (2 cores). A forma mais simples de fazer isso é comparar o valor de cada pixel com um limiar pré-definido: se o valor do pixel for igual ou superior ao limiar, ele vira branco ($v=\max$), caso contrário ele vira preto ($v=0$).

onde N é um limiar real entre 0.0 (0% do valor máximo) e 1.0 (100% do valor máximo). Caso o limiar não esteja definido, assume-se que seja 50%.

Exemplo de entrada e de saídas com limiar de 50% e 75%:

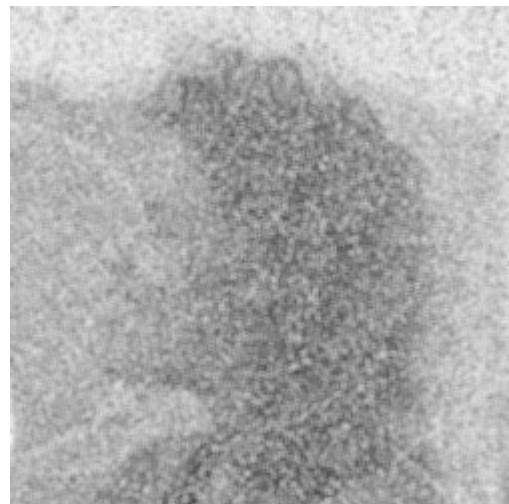


Filtros de redução de ruído pela média - Grupo 3

O filtro da média é usado para para “limpar” uma imagem, ou seja, reduzir o seu nível de ruído. O algoritmo é básico bem simples: para cada pixel, seu valor deve ser substituído pela média de todos os 9 pixels vizinhos (incluindo ele mesmo).

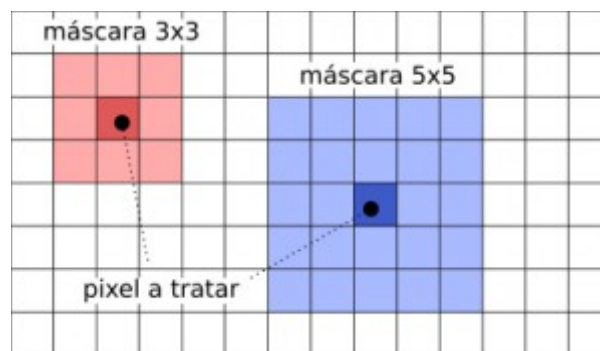
Deve ser tomado cuidado especial ao tratar os pixels nas primeiras e últimas linhas ou colunas, pois eles não têm todos os vizinhos. Nesses casos, devem ser considerados no cálculo somente os vizinhos existentes.

Exemplo de entrada e de saída:



Filtros da mediana - Grupo 4

O filtro da mediana reduz o nível de ruído de uma imagem sem prejudicar muito sua nitidez. Este filtro consiste basicamente em substituir o valor de um pixel pelo valor da mediana de seus pixels vizinhos. O número de vizinhos a considerar é definido por uma máscara, ou seja, a matriz de vizinhos que circunda o pixel a tratar (incluindo ele mesmo):



Requisitos:

O tamanho da máscara é um inteiro positivo ímpar, que pode ser informado na linha de comando (opção -m); caso não seja informado, o valor default é 3, para uma máscara de 3x3 pixels. Os pixels nas bordas da imagem não têm todos os vizinhos e portanto não devem ser filtrados. Para ordenar os valores dos pixels deve ser usada a função qsort da biblioteca C (man qsort).

Exemplo de entrada e de saída:



Filtros LBP (Local Binary Patterns) - Grupo 5

LBP (Local Binary Patterns) ou Padrões Binários Locais, é um descritor de textura muito utilizado em algoritmos de reconhecimento de imagens. O funcionamento desse filtro é simples: para cada pixel de uma imagem com 256 níveis de cinza, seleciona-se uma vizinhança de 8 pixels. O valor LBP é então calculado para o pixel central e armazenado na imagem destino, que possui as mesmas dimensões da imagem original.

Para o cálculo do LBP de um pixel, usa-se uma máscara com valores 2^n , com $n = \{0, \dots, 7\}$, ou seja: 1, 2, 4, 8, 16, 32, 64 e 128. O exemplo abaixo ilustra o cálculo de LBP para o pixel com valor 6:

6	5	2
7	6	1
9	3	7

(a)

1	0	0
1		0
1	0	1

(b)

1	2	4
8		16
32	64	128

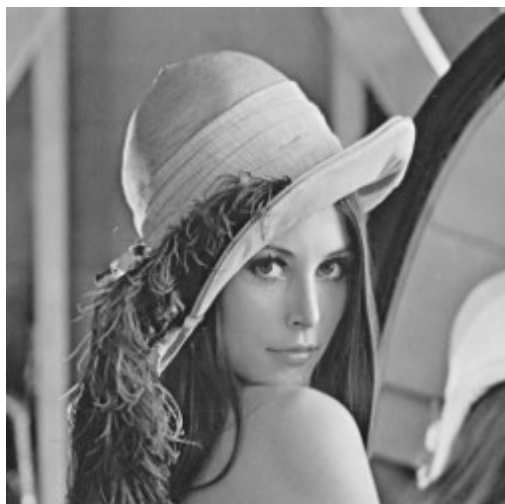
(c)

1	0	0
8		0
32	0	128

(d)

Considere o pixel central com valor 6 e seus oito vizinhos na imagem (a). O valor do pixel central é utilizado como limiar e todos os pixels com valor de intensidade maior ou igual a ele recebem 1, caso contrário, 0 (b). Esses valores são então multiplicados pela máscara (c), resultando então nos valores apresentados em (d). O valor LBP do pixel 6 é dado pela soma desses valores, ou seja $LBP = 1+8+32+128 = 169$. Note que essa codificação garante valores LBP entre 0 e 255.

Exemplo de entrada e de saída:



Outros Filtros (Mudança de Escala, Detecção de Bordas, Melhoria de Nitidez, Melhoria de Contraste) - Grupo 6, 7, n...

Instruções Gerais

- 1) O formato PGM permite definir imagens com pixels de 8 ou 16 bits. Para este trabalho, considere somente imagens com pixels de 8 bits (todas as imagens fornecidas como exemplo têm pixels de 8 bits).
- 2) Sempre que possível, as informações da imagem necessárias às funções devem ser transferidas como parâmetros (por valor ou por referência, dependendo da situação). Minimizar o uso de variáveis globais.
- 3) Use alocação dinâmica de memória para as imagens, para poder processar as imagens maiores. Só aloque a memória após encontrar o tamanho da imagem.
- 4) Para ser válido, o código deve **compilar e executar com sucesso**; códigos que não compilarem, não executarem até o fim, travarem, encerrarem com erros de sistema (segmentation fault, bus error, no more memory, etc) ou tiverem comportamento instável terão nota **zero**.
- 5) No final da tarefa estão previstas entrevistas individuais para defesa dos projetos realizados.