

Instituto Federal de Educação, Ciência e Tecnologia do Ceará

Engenharia da computação

Visão Computacional

Prof. Pedro Pedrosa

Relatório N^o 1 - Testes com manipulação de imagens

David Paulo Magalhães Araújo

Fortaleza

Março de 2018

Sumário

1	Introdução	2
2	Testes com filtros passa-baixa	2
2.1	Teste com filtro da mediana	2
2.2	Teste com filtro da média	3
2.3	Teste com filtro gaussiano	4
3	Testes com filtros passa-alta	5
3.1	Teste com filtro laplaciano	6
3.2	Teste com filtro prewit	7
3.3	Teste com filtro sobel	7
4	Testes com equalização	8
5	Testes com limiarização e multilimiarização	9
6	Resumo Geral	10
7	Referências Bibliográficas	11

1 Introdução

Este relatório tem como objetivo principal relatar os resultados dos testes realizados com filtros passa alta e passa baixa, bem como os resultados dos processos de equalização e limiarização, observando as mudanças na imagem e as razões para tais processos. A seguir, estão listados todos os processos que serão abordados nesse documento:

- Filtro da mediana
- Filtro da média
- Filtro gaussiano
- Filtro laplaciano
- Filtro sobel
- Filtro prewit
- Equalização de histograma
- Limiarização
- Multilimiarização

2 Testes com filtros passa-baixa

Nessa seção abordaremos os filtros da média, mediana e gaussiano.

Na implementação feita em C, o filtro da mediana pode ser feito como foi mostrado anteriormente selecionando a opção 0 (carregar imagem da Lenna) e depois a opção 9 (Filtro da mediana). Já os outros filtros passa-baixa são feitos com uma convolução (opção 6) na imagem carregada. Antes de fazer a convolução, o aplicativo solicitará por um kernel e, informando adequadamente, você consegue os filtros da média e gaussiano.

2.1 Teste com filtro da mediana

O filtro da mediana é o único filtro passa-baixa entre os três a serem abordados que não é uma convolução de um kernel sobre uma imagem. Ele consiste em alterar os pixels da imagem para a mediana de sua vizinhança, eliminando pixels que distoem muito de seus pixels vizinhos, sendo um de seus objetivos o de eliminar ruídos do tipo sal e pimenta.



(a) Antes do filtro

(b) Depois do filtro

Figura 1: Aplicação do filtro da mediana

Um dos efeitos dele é o de deixar a imagen suavizada, sem aparentes relevos, como um tecido.

2.2 Teste com filtro da média

O filtro da média é feito pela média aritmética de cada pixel de uma imagen e sua vizinhança. Um dos seus principais objetivos é o de remover ruídos, mas ele tem a desvantagem de não preservar tão bem as bordas ou fronteiras das imagens quando comparado ao filtro da mediana. Dado sua definição, O filtro da média pode ser implementando fazendo-se uma convolução da matriz de pixels da imagem com a seguinte máscara:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



(a) Antes do filtro

(b) Depois do filtro

Figura 2: Aplicação do filtro da média

2.3 Teste com filtro gaussiano

O filtro gaussiano é realizado pela convolução da imagem com um kernel com valores aproximados da função gaussiana. O filtro é usado para "borrar" a imagem e remover detalhes e ruídos, além de prover uma suavização mais gentil da imagem e uma preservação das bordas melhor do que o filtro da média. Para usar um filtro gaussiano, é necessário, primeiro, gerar um kernel com valores aproximados da função gaussiana, e depois fazer convolução desse com a imagem. A matrix, a seguir, é um exemplo de um kernel adequado:

$$\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$



(a) Antes do filtro

(b) Depois do filtro

Figura 3: Aplicação do filtro gaussiano

3 Testes com filtros passa-alta

Nessa seção abordaremos os filtros laplaciano, prewit e sobel.

Na implementação feita em C, É necessário, antes de qualquer operação, carregar uma imagem. Use a opção 0 para carregar a imagem da Lenna. No caso do filtro laplaciano, após carregar a imagem, faça uma convolução (opção 6) e forneça um kernel laplaciano, e o resultado será uma imagem aplicada com o filtro laplaciano. Para o prewit e o sobel, faça o seguinte passo-à-passo:

- Carregue a imagem, caso ainda não tenha feito (Opção 0 ou Opção 1)
- Faça uma convolução (opção 6) com o kernel horizontal do filtro
- Guarde a imagem em memória (opção 2)
- Carregue a imagem original novamente (Opção 0 ou Opção 1)
- Faça uma convolução (opção 6) com o kernel vertical do filtro
- Selecione a opção 3
- Selecione a opção da raiz da soma dos quadrados (opção 2)

O resultado será uma imagem com o filtro sobel ou prewit, dependendo dos kernels fornecidos nas convoluções

3.1 Teste com filtro laplaciano

A Laplaciana é uma medida isotropica 2-D da segunda derivada espacial de uma imagem. Ela destaca, na imagem, regiões em que haja rápida mudança de intensidade e é, portanto, frequentemente usada para detecção de bordas. a Laplaciana é frequentemente aplicada em uma imagem que foi antes suavizada com um filtro gaussiano para reduzir sua sensibilidade à ruídos.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



(a) Antes do filtro

(b) Depois do filtro

Figura 4: Aplicação do filtro laplaciano

3.2 Teste com filtro prewit

O filtro prewit é usado para detecção de bordas em uma imagem. Ele detecta dois tipos de bordas: horizontais e verticais. Bordas são calculadas através da diferença entre as intensidades dos pixels correspondentes de uma imagem. É possível somar o resultado horizontal e com o vertical e obter uma imagem destacando os dois tipos.

$$\left[\left(\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * \text{img} \right)^2 + \left(\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * \text{img} \right)^2 \right]^{\frac{1}{2}}$$

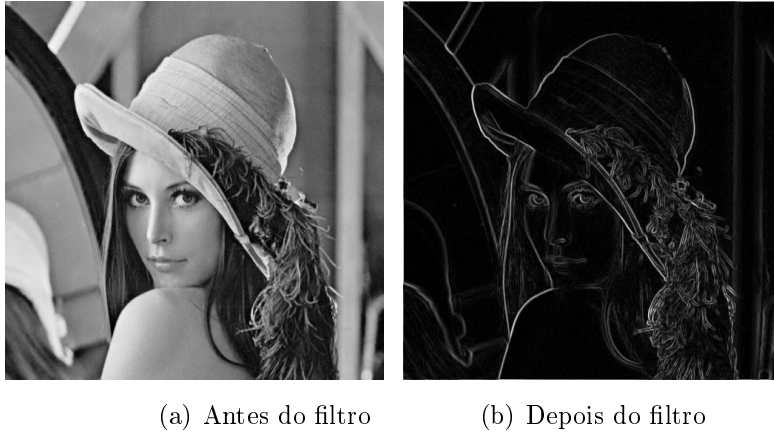


Figura 5: Aplicação do filtro prewit

3.3 Teste com filtro sobel

O filtro sobel é muito parecido com o filtro prewit. Ele também usa uma máscara derivadora, e é usado para detecção de bordas. Como o prewit, ele também pode detectar dois tipos de bordas: horizontais e verticais. A principal diferença é que no sobel, os coeficientes da máscara não são fixos e podem ser ajustadas livremente, desde que não viole nenhuma propriedade de uma máscara derivadora.

$$\left[\left(\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \text{img} \right)^2 + \left(\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \text{img} \right)^2 \right]^{\frac{1}{2}}$$



(a) Antes do filtro

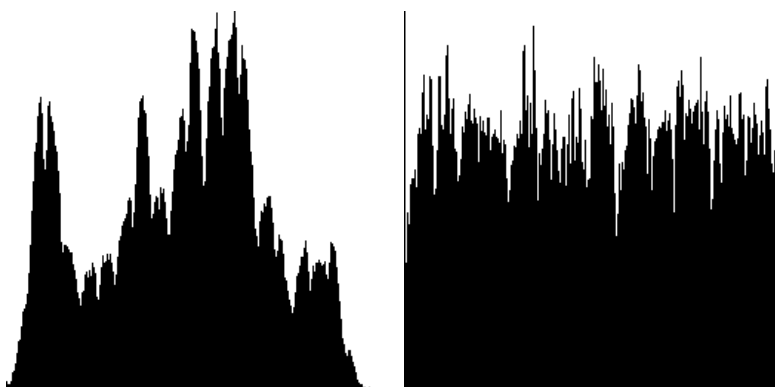
(b) Depois do filtro

Figura 6: Aplicação do filtro sobel

4 Testes com equalização

Nessa seção abordaremos os efeitos da equalização, tanto para o histograma da imagem quanto pra imagem em si.

A equalização consiste em redistribuir a frequência em que certas tonalidades de pixels aparecem na imagem, com o intuito de deixá-los o mais próximo possível. O efeito implicado da equalização é o aumento do contraste da imagem, devido a uma distribuição mais homogênea das cores ou tonalidades de cinza. O efeito no histograma é o de um gráfico mais suavizado e com menos oscilações do que o original.



(a) Antes

(b) Depois

Figura 7: Efeito da equalização no histograma da imagem

Para executar uma equalização de imagem na implementação em C, primeiro carregue uma imagem (opção 0 ou opção 1) e depois selecione equalização (opção 7)



(a) Antes

(b) Depois

Figura 8: Efeito da equalização na imagem

5 Testes com limiarização e multilimiarização

Nessa seção abordaremos os efeitos da limiarização e da multilimiarização na imagem

A limiarização consiste em transformar todas as tonalidades da imagem para branco ou preto, tomando como referência um pivô escolhido arbitrariamente. Ou seja, escolhido uma tonalidade central, todas as tonalidades menores do que essa serão transformadas em preto, e todas que forem iguais ou maiores à ela, se tornarão brancas. A Multilimiarização, por sua vez, divide todas as tonalidades em 3 ou mais, sendo elas branco, preto e outra(s) tonalidade(s) intermediária(s).



(a) Antes

(b) Depois

Figura 9: Efeito da limiarização na imagem



(a) Antes

(b) Depois

Figura 10: Efeito da multilimiarização na imagem

Para executar uma limiarização ou multilimiarização de imagem na implementação em C, primeiro carregue uma imagem (opção 0 ou opção 1) depois selecione limiarização (opção 8). Informe a quantidade de limiares, sendo 1 para limiarização e 2 ou mais para uma multilimiarização, depois disso será solicitado um pivô (ou mais) para realizar o processo escolhido.

6 Resumo Geral

Esse documento faz um breve relatório dos resultados obtidos pela implementação em C dos filtros e processos de manipulação de imagem abordados por ele. No decorrer do relatório, é possível notar que os resultados obtidos estão dentro do esperado para os testes realizados, vinculando teoria e prática como eram os objetivos do documento.

O documento ensina, também, como usar a implementação feita em C para verificar os testes que foram feitos e comparar com os resultados obtidos.

7 Referências Bibliográficas

Filtro da mediana (Em inglês):

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>

Filtro gaussiano (Em inglês):

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>

Filtro laplaciano (Em inglês):

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

Filtro sobel (Em inglês):

https://www.tutorialspoint.com/dip/sobel_operator.htm

Aula 3 da cadeira de visão computacional do IFCE:

<https://www.dropbox.com/s/yarq1zlgv5b9zwp/VC%20-%20Aula%203.pdf?dl=0>

Aula 4 da cadeira de visão computacional do IFCE:

<https://www.dropbox.com/s/y50pfpw16ueby0n/VC%20-%20Aula%204.pdf?dl=0>