Relatório - Trabalho 01

MO443 - Introdução ao Processamento de Imagem Digital

Vinicius Teixeira de Melo - RA: 230223

Universidade Estadual de Campinas
viniciusteixeira@liv.ic.unicamp.br

15 de Abril de 2019

I. Especificação do Problema

O objetivo deste trabalho é implementar alguns filtros de imagens no domínio espacial e de frequências. A filtragem aplicada a uma imagem digital é uma operação local que altera os valores de intensidade dos pixels da imagem levando-se em conta tanto o valor do pixel em questão quanto valores de pixels vizinhos.

No processo de filtragem, utiliza-se uma operação de convolução de uma máscara pela imagem. Este processo equivale a percorrer toda a imagem alterando seus valores conforme os pesos da máscara e as intensidades da imagem.

O trabalho está dividido em duas seções principais:

- Filtragem no Domínio Espacial
- Filtragem no Domínio de Frequências

Na seção de filtragem no domínio espacial, são dados 4 seguintes tipos de filtros:

(a) $h_1 =$

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(b) $h_2 = \frac{1}{256}$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

(c) $h_3 =$

-1	0	1
-2	0	2
-1	0	1

(d) $h_4 =$

-1	-2	-1
0	0	0
1	2	1

As aplicações dos filtros devem ser feitas de forma individual, porém, os filtros h_3 e h_4 devem ser aplicados de forma combinada, somando-se as respostas de cada um dos filtros por meio da seguinte expressão: $\sqrt{(h_3)^2 + (h_4)^2}$.

Na seção de filtragem no domínio de frequências, é necessário aplicar um filtro Gaussiano em uma imagem representada por seu espectro de Fourier, e a componente de frequência zero deve ser transladada para o centro do espectro. Nesse experimento, é necessário testar diferentes graus de suavização, de formaa borrar a imagem com mais ou menos intensidade.

II. Entrada de Dados

O código fonte criado para a execução de todas as tarefas está no notebook **Trabalho 01.ipynb**. O código foi criado para aceitar imagens em tons de cinza no formato RGB (*Red, Green and Blue*) do tipo PNG (*Portable Network Graphics*).

Para executar o notebook, basta iniciar o ambiente *Jupyter Notebook*, abrir o notebook **Trabalho 01.ipynb** e executar as células em ordem. Todo o algoritmo foi implementado na linguagem Python na versão 3.6.

As imagens de entrada utilizadas nos testes do algoritmo foram retiradas da página do prof. Hélio Pedrini: Imagens. Na pasta **imgs/** estão as duas imagens monocromáticas utilizadas nos testes: **baboon.png** e **butterfly.png**. As

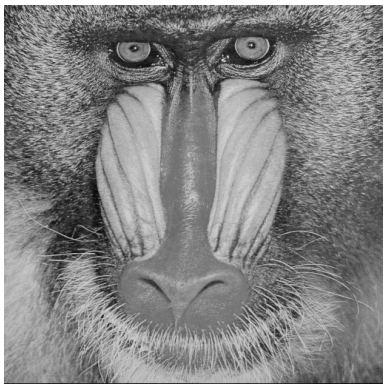

Figura 1: *baboon.png*

Figura 2: *butterfly.png*

dimensões das imagens de entrada utilizadas são 512x512.

III. Código e Definições

i. Preenchimento de borda

As imagens de entrada foram carregadas em memória utilizando uma função do **OpenCv** [1] que lê a imagem em escala de cinza e salva em uma estrutura I_{mxn}, na qual m e n representa a largura e a altura da imagem, respectivamente.

Para a aplicação da operação de convolução na imagem, foi necessário realizar um pré-processamento chamado de *padding*. A operação de *padding* é usada para gerar espaços em torno do conteúdo de um elemento, nesse caso, a imagem. Nesse trabalho, o *padding* gerado nas imagens foi de tamanho $\lfloor \frac{x}{2} \rfloor$, onde x é a dimensão do filtro que será convoluído com a imagem. Todo o *padding* foi preenchido com o valor 0, ou seja, com a cor preta.

ii. Convolução 2D

Convolução 2D é uma operação que, a partir de imagem e um filtro, resulta em uma terceira imagem que mede a soma do produto da imagem e o filtro inicial ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas.

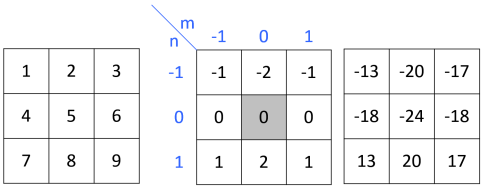

Figura 3: *Exemplo de imagem, filtro e resultado da convolução, respectivamente.*

A operação de convolução, primeiramente, rotaciona o filtro em 180° e depois multiplica *pixel* a *pixel*, da seguinte forma:

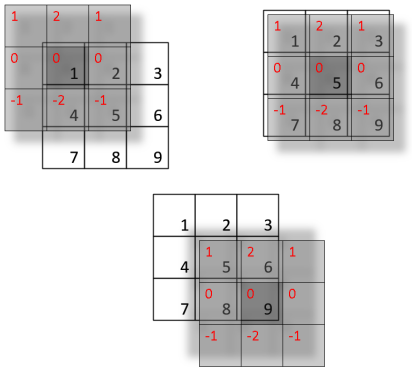

Figura 4: *Exemplo de aplicação de convolução 2D.*

Nesse trabalho, foram utilizados 2 tipos de filtros:

1. Filtro passa-alta: filtros utilizados para realçar certas características presentes nas

imagens, tais como bordas, linhas ou regiões de interesse [2].

2. Filtro passa-baixa: filtros utilizados para suavizar as imagens, uma vez que as frequências altas correspondem às transições abruptas são atenuadas [2].

Os filtro h_1, h_3 e h_4 são exemplos de filtros passa-alta. Já o filtro h_2 é um exemplo de filtro passa-baixa. A seguir temos a exemplificação gráfica desses filtros.

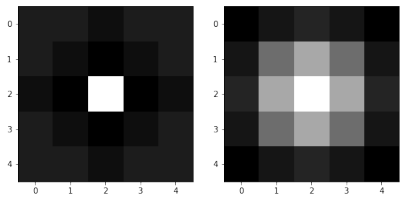

Figura 5: *Filtros h_1 e h_2.*

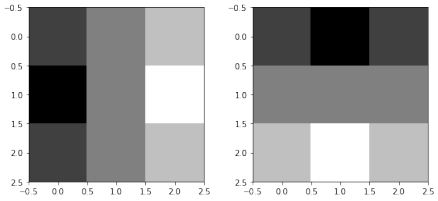

Figura 6: *Filtros h_3 e h_4.*

iii. Redimensionamento

A operação de redimensionamento foi utilizada para transformar as saídas da operação de convolução 2D, que não necessariamente estavam no intervalo de escalas de cinza de $[0, 255]$, de volta ao intervalo de escalas de cinza representado por inteiros de 8 *bits*.

iv. Transformada Discreta de Fourier

A Transformada Discreta de Fourier (DFT) é uma transformação de coordenadas em componentes pertencentes ao conjunto dos números complexos, em que cada coeficiente é obtido pela combinação linear dos elementos da entrada com o núcleo da transformada [2].

A Transformada de Fourier para duas variáveis pode ser escrita da seguinte forma:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) exp\left[-j2\pi(ux+vy)\right] dxdy$$

e a partir de $F(u,v)$, pode-se obter $f(x,y)$ através da Transformada Inversa de Fourier:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) exp\left[j2\pi(ux+vy)\right] dudv.$$

Na filtragem no domínio de frequência, o principal objetivo, é que as operações de convolução possuem o custo computacional menor do que no domínio espacial. Portanto, normalmente, o processamento de imagens possuem a seguinte sequência.

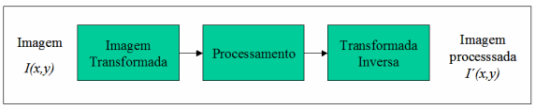

Figura 7: *Sequência usualmente utilizada para as operações de processamento de imagens.*

v. Filtro Gaussiano

Os filtros Gaussianos são um tipo de filtro passa-baixa, os quais suavizam a imagem filtrada. Nos filtros Gaussianos, os coeficientes da máscara são derivados a partir de uma função Gaussiana bidimensional [2]. A função Gaussiana discreta com média zero e desvio padrão σ é definida como

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right).$$

As variações de sigma utilizados neste trabalho foram: $\sigma = [2, 4, 8, 16, 32, 64]$.

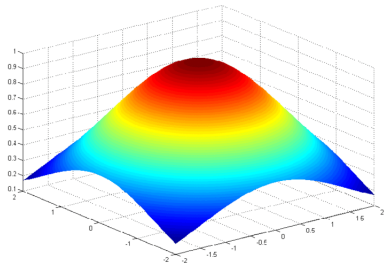

Figura 8: *Exemplificação gráfica de um filtro Gaussiano.*

IV. Saída de Dados

Os imagens resultantes da seção **1.1** e **1.2** foram salvas dentro da pasta **resultados/** utilizando uma função da biblioteca **matplotlib** chamada **matplotlib.pyplot.imsave()** [3].

O formato dos nomes de saída estão da seguinte forma: para a seção **1.1** o padrão é o nome da imagem de entrada (*baboon* ou *butterfly*) concatenado com o nome do filtro aplicado sobre ela.

V. Resultados

Referências

[1] Welcome to opencv documentation! https://docs.opencv.org/2.4/index.html Acesso em: 15/04/2019.

[2] Pedrini, Hélio, and William Robson Schwartz. Análise de imagens digitais: princípios, algoritmos e aplicações. Thomson Learning, 2008.

[3] Matplotlib Version 3.0.3 https://matplotlib.org/contents.html Acesso em: 15/04/2019.