

Relatório Trabalho 2 -

Renato Alberto dos Santos¹, Rodrigo Norio Tshako²

¹Departamento de Informática - Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

ra96565@uem.br¹, ra95376@uem.br²

Resumo. *No processo digital de imagem existe várias maneiras de se aumentar a nitidez de uma imagem. O processo high-boost é um exemplo de filtro de nitidez que realiza o aguçamento de imagens digitais. O presente trabalho tem como objetivo o desenvolvimento de um filtro high-boost com duas abordagens diferentes. Os resultados mostram que, para a imagem de entrada escolhida, as duas versões do high-boost atingiram bons resultados.*

Palavras-chave: *Processamento digital de imagem, High-Boost, Fourier.*

1. Introdução

Uma imagem digital pode ser descrita por coordenadas bidimensionais onde seus valores são discretos e finitos. O processamento digital de imagem pode ser caracterizado por qualquer meio de processamento de dados em que os parâmetros de entrada e saída sejam imagens digitais.

Neste contexto, os filtros de imagens são técnicas que manipulam os dados na imagem digital de entrada, gerando uma imagem digital como saída. Na aplicação de um filtro, os dados de entrada são modificados mantendo a essência da imagem de origem.

Os filtros podem ser aplicados no domínio espacial ou domínio da frequência. O domínio espacial é representado por meio de uma matriz bidimensional, onde cada pixel representa uma posição na matriz. Enquanto, o domínio da frequência é caracterizado por meio de sinais digitais.

O filtro *high-boost* é um filtro utilizado para o aguçamento de imagens, ou seja, esse filtro melhora a nitidez das imagens e por isso é muito utilizado na indústria gráfica. É uma técnica simples e pode ser aplicada tanto no domínio espacial quanto no domínio da frequência.

O presente trabalho aplica duas versões do filtro *high-boost*, uma versão com borramento no domínio espacial e outra versão com borramento no domínio da frequência, em uma imagem com tons de cinza.

2. Domínio Espacial

Uma imagem digital pode ser representada por uma matriz bidimensional em que cada posição representa um pixel da imagem. O domínio espacial consiste em técnicas que atuam diretamente sobre a matriz de da imagem digitalizada, em que, é manipulado uma vizinhança de pixels afim de conseguir o efeito desejado [Filho and Neto 1999].

Os filtros no domínio espacial podem ser representados por meio de máscaras e cada uma delas possuem suas definições matemáticas. Entretanto, é possível representar essas definições matemáticas através de aproximações em forma de matrizes e por meio da convolução, aplicá-las na matriz da imagem [Gonzalez and Woods 2010].

3. Domínio da Frequência

Na matemática, as funções periódicas podem ser representadas por meio da soma da função seno ou da função cosseno em diferentes frequências, cada uma multiplicada por um coeficiente diferente. O que torna esse conceito interessante é que a função original no domínio da frequência pode ser reconstruída para seu domínio original sem perder informações durante o processo [de Albuquerque 2013].

Inicialmente o domínio da frequência era usado em aplicações industriais para difusão de calor. Entretanto, com o surgimento do computador foi possível utilizar os conceitos do domínio da frequência para analisar sinais digitais. Nesse contexto, é possível aplicar a transformada de Fourier para converter uma imagem digital em um sinal digital, tornando possível manipular esse sinal e voltar ele para seu domínio original sem perder informações [Gonzalez and Woods 2010].

4. High Boost

O *high-boost* é um filtro que proporciona o aumento de nitidez das imagens, em outras palavras, ele realiza o aguçamento das imagens. Esse processo pode ser chamado de máscara de nitidez e consiste nos seguintes passos:

1. Aplicar filtro de borramento
2. Calcular a diferença entre imagem original e imagem borrada
3. Adicionar a máscara (resultado da etapa anterior) na imagem original

Matematicamente a mascara é representada pela função:

$$G_{mascara}(x, y) = f(x, y) - \bar{f}(x, y) \quad (1)$$

Onde $\bar{f}(x, y)$ é a imagem borrada.

Depois de obter a máscara, o próximo passo é adicionar a máscara na imagem original:

$$HighBoost(x, y) = f(x, y) + k * G_{mascara}(x, y) \quad (2)$$

Onde k é uma constante maior que 1. O valor de k depende de acordo com a imagem, vale ressaltar que um k muito elevado pode gerar resultados altos, tornando necessário a normalização dos resultados [Gonzalez and Woods 2010].

5. Gaussian Blur

Os filtros de suavização são utilizados para aplicar borramento e redução de ruídos das imagens. São classificados como filtros de passa-baixa e podem ser aplicados no domínio espacial e domínio da frequência [Filho and Neto 1999].

O filtro gaussiano unidimensional é representado por:

$$Gaussian(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

Enquanto o filtro gaussiano bidimensional é dado pela função a seguir:

$$Gaussian(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4)$$

Em aplicações no domínio espacial o filtro gaussiano pode ser classificado como filtro da média e pode ser representado por meio de uma matriz [Gonzalez and Woods 2010].

6. Transformada de Fourier

O matemático Jean Baptiste Joseph Fourier descobriu que funções periódicas podem ser apresentadas como a soma da função seno ou função cossenos de diferentes frequências, cada uma multiplicada por um coeficiente diferente. Além disso, representou as funções não periódicas como a integral de seno ou cosseno multiplicada por uma função de ponderação, e nesse contexto surgiu a transformada de Fourier [de Albuquerque 2013].

A transformada de Fourier permite que uma função possa ser manipulada no domínio da frequência e reconstruída em seu domínio original, por meio de um processo inverso, sem perder informações durante o processo. Com a chegada do computador, foi possível aplicar esse conceito em imagens digitais [Gonzalez and Woods 2010]

7. Metodologia

Para a execução do trabalho foi utilizado como base as seguintes especificações.

7.1. Hardware e Software

O trabalho foi desenvolvido e testado em uma máquina com as seguintes especificações:

- Processador Intel Core I5-4210, com frequência de 1,7 GHz;
- 8G de memória RAM;
- HD de 1TB;
- Sistema operacional Windows 10 versão 1909.
- Editor de texto Visual Studio Code
- Linguagem Python 3.8.5

7.2. Desenvolvimento

O presente trabalho consiste em desenvolver o filtro *high-boost* no domínio espacial e no domínio da frequência, por isso o presente estudo foi realizado em duas partes.

O filtro *high-boost* foi aplicado em uma imagem de entrada em tons de cinza.



Figura 1. Imagem de entrada

A figura 1 representa os dados de entrada com dimensões de 320×138 .

Como dito em seções anteriores, o algoritmo do *high-boost* é bem simples e consiste nos seguintes passos:

1. Aplicar filtro de borramento
2. Calcular a diferença entre imagem original e imagem borrada
3. Adicionar a máscara (resultado da etapa anterior) na imagem original

A parte 1 do presente trabalho aplica o filtro gaussiano no domínio espacial para borrar a imagem e a parte 2 aplica a transformada de Fourier e filtro gaussiano no domínio da frequência para borrar a imagem.

8. Resultados

Nesta seção serão apresentados os resultados obtidos da parte 1 e 2.

8.1. Parte 1 - Domínio Espacial

A primeira etapa desta parte foi realizar o borramento da imagem por meio do filtro gaussiano no domínio espacial:



Figura 2. Imagem Borrada

A figura 2 mostra o resultado do borramento no domínio espacial.

Com a imagem borrada, foi possível calcular a diferença entre imagem de entrada e imagem borrada, resultando na máscara de nitidez (representada pela fórmula (1) da seção 4):



Figura 3. Mascara de nitidez

Como a imagem de entrada e a imagem borrada possuem tons de cinza semelhantes, a diferença entre elas gera resultados próximos a zero (cor preta), como mostra a máscara na figura 3.

A última etapa é descrita pela fórmula (2) (seção 4), em que a máscara é adicionada à imagem de entrada:

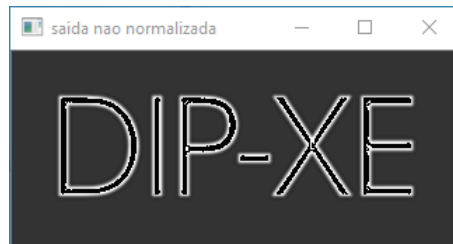


Figura 4. Imagem de Saída não Normalizada

É possível observar na figura 4 que o resultado não normalizado gera uma imagem inesperada, com cores pretas em lugares que deveriam ser branco.

Aplicando a normalização dos resultados, é gerado uma imagem com maior nitidez:



Figura 5. Imagem de Saída Normalizada

É importante ressaltar que foi utilizado $k = 2$ para produzir o resultado mostrado na figura 5.

8.2. Parte 2 - Domínio da Frequência

A primeira etapa da parte 2 foi aplicar o borramento da imagem no domínio da frequência, e para isso foi utilizado a transformada de Fourier para converter a imagem de entrada e o filtro gaussiano em frequências. Em seguida foi realizado a convolução das frequências e por meio da transformada de Fourier inversa foi possível voltar os resultados para o domínio de origem.

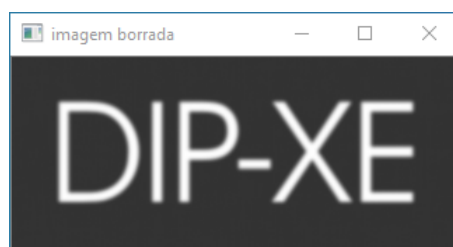


Figura 6. Imagem Borrada

A figura 6 ilustra a imagem borrada no domínio da frequência.

Subtraindo a imagem de entrada pela imagem borrada foi possível obter a máscara de nitidez:



Figura 7. Mascara de nitidez

É possível observar que a figura 7 também apresenta bastante cor preta, indicando que a subtração gerou resultados bem próximos a 0.

Adicionando a máscara na imagem de entrada foi obtido a imagem de saída:



Figura 8. Imagem de Saída não Normalizada

É possível observar na figura 8 que a imagem de saída sem normalização gera resultados indesejados, com cores pretas onde deveriam ser branco.

Normalizando os resultados, foi obtido uma imagem com maior nitidez:

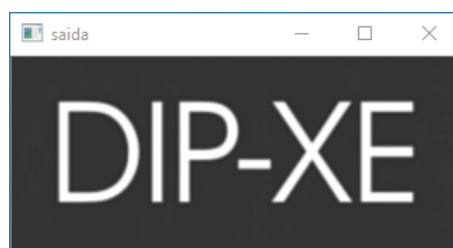


Figura 9. Imagem de Saída Normalizada

O resultado obtido representado pela figura 9 foi alcançado com valor de $k = 2$.

9. Conclusão

Em algumas aplicações de processamento de imagens digitais, torna-se necessário a utilização de técnicas de aguçamento de imagem afim de melhorar a nitidez dela. O *high-boost* é um filtro de aguçamento que possui aplicações tanto no domínio espacial quanto no domínio da frequência.

Considera-se que o presente trabalho atendeu os objetivos propostos, uma vez que, foi realizado o desenvolvimento do filtro *high-boost* em duas abordagens diferentes, uma com o borramento da imagem no domínio espacial e outra com o borramento da imagem no domínio da frequência.

Com base nos resultados obtidos, é possível observar que as duas abordagens melhoraram a nitidez da imagem final. Além disso, nota-se que em ambas as abordagens foram utilizadas a máscara gaussiana para borrar a imagem de entrada, mas vale ressaltar que uma versão foi feita no domínio espacial e a outra versão no domínio da frequência.

Portanto, é possível concluir que o domínio em que é realizado o borramento da imagem não foi um fator que impactou bruscamente o resultado final, uma vez que as duas versões apresentaram resultados parecidos para o filtro *high-boost*. Logo, as duas abordagens são excelentes para o aguçamento de uma imagem em tons de cinza.

10. Implementação

O código a seguir foi desenvolvido em Python 3.8.5 utilizando as bibliotecas numpy e cv2.

10.1. Parte 1 - Domínio Espacial

```
import numpy as np
import cv2

def carregarimagem():
    imagem = input('Digite o nome da imagem com sua extensão: ')
    img = cv2.imread(imagem, 0)
    matimg = np.array(img)

    return matimg, img

def showimagemcv(img, imgblur, mascara, saida, naonormalizado):
    cv2.imshow('original', img)
    cv2.imshow('imagem borrada', imgblur)
    cv2.imshow('mascara', mascara)
    cv2.imshow('saida nao normalizada', naonormalizado)
    cv2.imshow('saida', saida)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

matimg, img = carregarimagem()

#parte 1
```

```

imgblur = cv2.GaussianBlur(img, (3,3), 1)

mascara = np.int32(matimg) - np.int32(imgblur)

mascaral = 2*np.int32(mascara)

saida = (np.int32(img) + mascaral)

naonormalizado = saida
naonormalizado = np.uint8(naonormalizado)

normalizamenor = saida < 0
saida[normalizamenor] = 0
normalizamaior = saida > 255
saida[normalizamaior] = 255

saida = np.uint8(saida)

img = np.uint8(img)
mascaral = np.uint8(mascaral)
imgblur = np.uint8(imgblur)

showimagemcv(img, imgblur, mascaral, saida, naonormalizado)

```

10.2. Parte 2 - Domínio da Frequência

```

import numpy as np
import cv2

def carregarimagem():
    imagem = input('Digite o nome da imagem com sua extensão: ')
    img = cv2.imread(imagem, 0)
    matimg = np.array(img)

    return matimg, img

def showimagemcv(img, imgblur, mascara, saida, naonormalizado):
    cv2.imshow('original', img)
    cv2.imshow('imagem borrada', imgblur)
    cv2.imshow('mascara', mascara)
    cv2.imshow('saida nao normalizada', naonormalizado)
    cv2.imshow('saida', saida)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

```



```

matimg, img = carregarimagem()

kernel = np.ones([3, 3], dtype=np.uint8)
padding = np.zeros([len(matimg), len(matimg[0])], dtype='f')

altura = int(len(padding)/2)
largura = int(len(padding[0])/2)

shape = np.maximum(kernel.shape, padding.shape)
h = np.zeros((shape), dtype=np.float)
h[:padding.shape[0], :padding.shape[1]] = padding

h[altura][largura] = 4/16
h[altura-1][largura-1] = 1/16
h[altura+1][largura+1] = 1/16
h[altura+1][largura-1] = 1/16
h[altura-1][largura+1] = 1/16
h[altura][largura-1] = 2/16
h[altura][largura+1] = 2/16
h[altura+1][largura] = 2/16
h[altura-1][largura] = 2/16

#parte 2

fourier = np.fft.fft2(matimg)

h = np.absolute(np.fft.fft2(h))

fourier = fourier * h

fourier = np.absolute(np.fft.ifft2(fourier))

```

```
maskara = np.int32(mating) - np.int32(fourier)

maskara1 = 2*np.int32(maskara)

saida = (np.int32(img) + maskara1)
naonormalizado = saida
naonormalizado = np.uint8(naonormalizado)

normalizamenor = saida < 0
saida[normalizamenor] = 0
normalizamaior = saida > 255
saida[normalizamaior] = 255

saida = np.uint8(saida)

img = np.uint8(img)
maskara1 = np.uint8(maskara1)
fourier = np.uint8(fourier)

showimagemcv(img, fourier, maskara1, saida, naonormalizado)
```

Referências

- de Albuquerque, H. R. (2013). *Fusão de imagens no domínio da frequência baseada em foco*. Universidade Federal de Pernambuco.
- Filho, O. M. and Neto, H. V. (1999). *Processamento Digital de Imagens*. BrasPort.
- Gonzalez, R. and Woods, R. (2010). *Processamento Digital de Imagens*, volume 3. PE-ARSON.